

# QUASIRANDOM LOAD BALANCING\*

TOBIAS FRIEDRICH<sup>†</sup>, MARTIN GAIRING<sup>‡</sup>, AND THOMAS SAUERWALD<sup>§</sup>

**Abstract.** We propose a simple distributed algorithm for balancing indivisible tokens on graphs. The algorithm is completely deterministic, though it tries to imitate (and enhance) a random algorithm by keeping the accumulated rounding errors as small as possible.

Our new algorithm surprisingly closely approximates the idealized process (where the tokens are divisible) on important network topologies. On  $d$ -dimensional torus graphs with  $n$  nodes it deviates from the idealized process only by an additive constant. In contrast to that, the randomized rounding approach of Friedrich and Sauerwald [11] can deviate up to  $\Omega(\text{polylog}(n))$  and the deterministic algorithm of Rabani, Sinclair and Wanka [32] has a deviation of  $\Omega(n^{1/d})$ . This makes our quasirandom algorithm the first known algorithm for this setting which is optimal both in time and achieved smoothness. We further show that also on the hypercube our algorithm has a smaller deviation from the idealized process than the previous algorithms.

**1. Introduction.** Load balancing is an important requisite for the efficient utilization of computational resources in parallel and distributed systems. The aim is to reallocate the load such that at the end each node has approximately the same load. Load balancing problems have various applications, *e.g.*, for scheduling [36], routing [5], and numerical computation [37, 38].

Typically, load balancing algorithms iteratively exchange load along edges of an undirected connected graph. In the natural *diffusion paradigm*, an arbitrary amount of load can be sent along each edge at each step [30, 32]. For the *idealized* case of divisible load, a popular diffusion algorithm is the first-order-scheme by Subramanian and Scherson [35] whose convergence rate is fairly well captured in terms of the spectral gap [26].

However, for many applications the assumption of divisible load may be invalid. Therefore, we consider the *discrete* case where the load can only be decomposed in indivisible unit-size tokens. It is a very natural question by how much this integrality assumption decreases the efficiency of load balancing. In fact, finding a precise quantitative relationship between the discrete and the idealized case is an open problem posed by many authors, *e.g.*, [9, 11, 14, 15, 27, 30, 32, 35].

A simple method for approximating the idealized process was analyzed by Rabani, Sinclair, and Wanka [32]. Their approach (which we will call “RSW-algorithm”) is to round down the fractional flow of the idealized process. They introduce a very useful parameter of the graph called *local divergence* and prove that it gives tight upper bounds on the deviation between the idealized process and their discrete process. However, one drawback of the RSW-algorithm is that it can end up in rather unbalanced states (cf. Proposition 6.1). To overcome this problem, Friedrich and Sauerwald analyzed a new algorithm based on randomized rounding [11]. On many graphs, this algorithm approximates the idealized case much better than the approach of always rounding down of the RSW-algorithm. A natural question is whether this randomized algorithm can be derandomized without sacrificing on its performance. For the graphs considered in this work, we answer this question to the positive. We introduce a *quasirandom load balancing algorithm* which rounds up or down deterministically such that the accumulated rounding errors on each edge are minimized. Our approach follows the concept of quasirandomness as it deterministically imitates the expected behavior of its random counterpart.

\*A preliminary conference version [12] appeared in the 21st ACM-SIAM Symposium on Discrete Algorithms (SODA 2010).

<sup>†</sup>Max-Planck-Institut für Informatik, Saarbrücken, Germany.

<sup>‡</sup>University of Liverpool, Liverpool, United Kingdom.

<sup>§</sup>Simon Fraser University, Burnaby, Canada.

*Our Results.* We focus on two network topologies: hypercubes and torus graphs. Both have been intensively studied in the context of load balancing (see *e.g.*, [11, 13, 19, 31, 32]). We measure the smoothness of the load by the so-called *discrepancy* (see *e.g.* [9, 11, 15, 32]) which is the difference between the maximum and minimum load among all nodes.

For *d-dimensional torus graphs* we prove that our quasirandom algorithm approximates the idealized process up to an additive constant (Theorem 5.4). More precisely, for all initial load distributions and time steps, the load of any vertex in the discrete process differs from the respective load in the idealized process only by a constant. This holds even for non-uniform torus graphs with different side-lengths (cf. Definition 5.1). For the uniform torus graph our results are to be compared with a deviation of  $\Omega(\text{polylog}(n))$  for the randomized rounding approach (Theorem 6.3) and  $\Omega(n^{1/d})$  for the RSW-algorithm (Proposition 6.1). Hence despite our approach is deterministic, it also improves over its random counterpart. Starting with an initial discrepancy of  $K$ , the idealized process reaches a constant discrepancy after  $\mathcal{O}(n^{2/d} \log(Kn))$  steps (cf. Corollary 3.2). Hence the same holds for our quasirandom algorithm, which makes it the first algorithm for the discrete case which is optimal both in time and discrepancy.

For the *hypercube* we prove a deviation of our quasirandom algorithm from the idealized process of  $\Theta(\log n)$  (Theorem 4.2). For this topology we also show that the deviation of the random approach is  $\Omega(\log n)$  (Theorem 6.2) while the deviation of the RSW-algorithm is  $\Omega(\log^2 n)$  (Proposition 6.1). Again, our quasirandom algorithm is at least as good as the randomized rounding algorithm and substantially better than the RSW-algorithm [32]. In particular, we obtain that for any load vector with initial discrepancy  $K$ , our quasirandom algorithm achieves a discrepancy of at most  $\log n$  after  $\mathcal{O}(\log n \log(Kn))$  steps.

*Our Techniques.* Instead of analyzing our quasirandom algorithm directly, we examine a new generic class of load balancing algorithms that we call *bounded error diffusion* (BED). Roughly speaking, in a BED algorithm the *accumulated* rounding error on each edge is bounded by some constant at all times. This class includes our quasirandom algorithm.

The starting point of [32] and [11] as well as our paper is to express the deviation from the idealized case by a certain sum of weighted rounding errors (equation (3.1)). In this sum, the rounding errors are weighted by transition probabilities of a certain random walk. Roughly speaking, Rabani et al. [32] estimate this sum directly by adding up all transition probabilities. In the randomized approach of [11], the sum is bounded by Chernoff-type inequalities relying on independent rounding decisions. We take a completely different approach and prove that the transition probabilities between two fixed vertices are unimodal in time (cf. Theorem 4.9 for the hypercube). This allows us to upper bound the complete sum by its maximal summand (Lemma 3.6) for BED algorithms. The intriguing combinatorial property of *unimodality* is the heart of our proof and seems to be the main reason why we can outperform the previous approaches. Even though unimodality has a one-line definition, it has become apparent that proving it can be a very challenging task requiring intricate combinatorial constructions or refined mathematical tools (see *e.g.* Stanley’s survey [34]).

It turns out that this is also the case for the considered transition probabilities of torus graphs and hypercubes. The reason is that explicit formulas seem to be intractable and typical approximations (*e.g.* Poissonization [6]) are way too loose to compare consecutive transition probabilities. For the *d-dimensional torus*, we use a local central limit theorem to approximate the transition probabilities by a multivariate normal distribution which is known to be unimodal.

On hypercubes the above method fails as several inequalities for the torus graph

are only true for constant  $d$ . However, we can employ the additional symmetries of the hypercube to prove unimodality of the transition probabilities by relating it to a random walk on a weighted path. Somewhat surprisingly, this intriguing property was unknown before, although random walks on hypercubes have been intensively studied (see *e.g.* [6, 21, 28]).

We prove this unimodality result by establishing an interesting result concerning first-passage probabilities of a random walk on paths with arbitrary transition probabilities: If the loop probabilities are at least  $1/2$ , then the first-passage probability distribution can be expressed as a convolution of independent geometric distributions. In particular, this implies that these probabilities are log-concave. Reducing the random walk on a hypercube to a random walk on a weighted path, we obtain that the transition probabilities on the hypercube are unimodal. Estimating the maximum probabilities via a balls-and-bins-process, we finally obtain our upper bound on the deviation for the hypercube.

We believe that our probabilistic result for paths is of independent interest, as random walks on the paths are among the most extensively studied stochastic processes. Moreover, many analyses of randomized algorithms can be reduced to such random walks (see *e.g.* [29, Thm. 6.1]).

*Related Work.* In the approach of Elsässer and Sauerwald [8] certain interacting random walks are used to reduce the load deviation. This randomized algorithm achieves a constant additive error between the maximum and average load on hypercubes and torus graphs in time  $\mathcal{O}(\log(Kn)/(1 - \lambda_2))$ , where  $\lambda_2$  is the second largest eigenvalue of the diffusion matrix. However, in contrast to our deterministic algorithm, this algorithm is less natural and more complicated (*e.g.*, the nodes must know an accurate estimate of the average load).

Aiello et al. [1] and Ghosh et al. [15] studied balancing algorithms where in each time step at most one token is transmitted over each edge. Due to this restriction, these algorithms take substantially more time, *i.e.*, they run in time at least linear in the initial discrepancy  $K$ . Nonetheless, the best known bounds on the discrepancy are only polynomial in  $n$  for the torus and  $\Omega(\log^5 n)$  for the hypercube [15].

In another common model, nodes are only allowed to exchange load with at most one neighbor in each time step, see *e.g.*, [11, 14, 32]. In fact, the afore-mentioned randomized rounding approach [11] was analyzed in this model. However, the idea of randomly rounding the fractional flow such that the expected error is zero naturally extends to our diffusive setting where a node may exchange load with all neighbors simultaneously.

*Quasirandomness* describes a deterministic process which imitates certain properties of a random process. Our quasirandom load balancing algorithm imitates the property that rounding up and down the flow between two vertices occurs roughly equally often by a deterministic process which minimizes these rounding errors directly. This way, we keep the desired property that the “expected” accumulated rounding error is zero, but remove almost all of its (undesired) variance. Similar concepts have been used for deterministic random walks [4], external mergesort [2], and quasirandom rumor spreading [7]. The latter work presents a quasirandom algorithm which is able to broadcast a piece of information at least as fast as its random counterpart on the hypercube and most random graphs. However, in case of rumor spreading the quasirandom protocol is just slightly faster than the random protocol while the quasirandom load-balancing algorithm presented here substantially outperforms its random counterpart.

*Organization of the paper.* In Section 2, we give a description of our bounded error diffusion (BED) model. For a better comparison, we present some results for the previous algorithms of [11] and [32] in Section 6. In Section 3, we introduce our basic method which is used in Sections 4 and 5 to analyze BED algorithms on hypercubes and torus

graphs, respectively.

**2. Model and algorithms.** We aim at balancing load on a connected, undirected graph  $G = (V, E)$ . Denote by  $\deg(i)$  the *degree* of node  $i \in V$  and let  $\Delta = \Delta(G) = \max_{i \in V} \deg(i)$  be the maximum degree of  $G$ . The balancing process is governed by an ergodic, doubly-stochastic diffusion matrix  $\mathbf{P}$  with

$$\mathbf{P}_{i,j} = \begin{cases} \frac{1}{2\Delta} & \text{if } \{i, j\} \in E, \\ 1 - \frac{\deg(i)}{2\Delta} & \text{if } i = j, \\ 0 & \text{otherwise.} \end{cases}$$

Let  $x^{(t)}$  be the load-vector of the vertices at step  $t$  (or more precisely, after the completion of the balancing procedure at step  $t$ ). The *discrepancy* of such a (row) vector  $x$  is  $\max_{i,j} (x_i - x_j)$ , and the discrepancy at step 0 is called initial discrepancy  $K$ .

*The idealized process.* In one time step each pair  $(i, j)$  of adjacent vertices shifts divisible tokens between  $i$  and  $j$ . We have the following iteration,  $x^{(t)} = x^{(t-1)}\mathbf{P}$  and inductively,  $x^{(t)} = x^{(0)}\mathbf{P}^t$ . Equivalently, for any edge  $\{i, j\} \in E$  and step  $t$ , the flow from  $i$  to  $j$  at step  $t$  is  $\mathbf{P}_{i,j}x_i^{(t-1)} - \mathbf{P}_{j,i}x_j^{(t-1)}$ . Note that the symmetry of  $\mathbf{P}$  implies that for  $t \rightarrow \infty$ ,  $x^{(t)}$  converges towards the uniform vector  $(1/n, 1/n, \dots, 1/n)$ .

*The discrete process.* There are different ways how to handle non-divisible tokens. We define the following *bounded error diffusion* (BED) model. Let  $\Phi_{i,j}^{(t)}$  denote the integral flow from  $i$  to  $j$  at time  $t$ . As  $\Phi_{i,j}^{(t)} = -\Phi_{j,i}^{(t)}$ , we have  $x_i^{(t)} = x_i^{(t-1)} - \sum_{j: \{i,j\} \in E} \Phi_{i,j}^{(t)}$ . Let  $e_{i,j}^{(t)} := (\mathbf{P}_{i,j}x_i^{(t-1)} - \mathbf{P}_{j,i}x_j^{(t-1)}) - \Phi_{i,j}^{(t)}$  be the excess load (or lack of load) allocated to  $i$  as a result of rounding on edge  $\{i, j\}$  in time step  $t$ . Note that for all vertices  $i$ ,  $x_i^{(t)} = (x^{(t-1)}\mathbf{P})_i + \sum_{j: \{i,j\} \in E} e_{i,j}^{(t)}$ . Let now  $\Lambda$  be an upper bound for the accumulated rounding errors (deviation from the idealized process), that is,  $|\sum_{s=1}^t e_{i,j}^{(s)}| \leq \Lambda$  for all  $t \in \mathbb{N}$  and  $\{i, j\} \in E$ . All our bounds still hold if  $\Lambda$  is a function of  $n$  and/or  $t$ , but we only say that an algorithm is a *BED algorithm* if  $\Lambda$  is a constant.

Our new *quasirandom diffusion algorithm* chooses for  $\mathbf{P}_{i,j}x_i^{(t)} \geq \mathbf{P}_{j,i}x_j^{(t)}$  the flow  $\Phi_{i,j}^{(t)}$  from  $i$  to  $j$  to be either  $\Phi_{i,j}^{(t)} = \lfloor \mathbf{P}_{i,j}x_i^{(t)} - \mathbf{P}_{j,i}x_j^{(t)} \rfloor$  or  $\Phi_{i,j}^{(t)} = \lceil \mathbf{P}_{i,j}x_i^{(t)} - \mathbf{P}_{j,i}x_j^{(t)} \rceil$  such that  $|\sum_{s=1}^t e_{i,j}^{(s)}|$  is minimized. This yields a BED algorithm with  $\Lambda \leq 1/2$  and can be implemented with  $\lceil \log_2 \Delta \rceil$  storage per edge. Note that one can imagine various other natural (deterministic or randomized) BED algorithms. To do so, the algorithm only has to ensure that the errors do not add up to more than a constant.

With above notation, the *RSW-algorithm* uses  $\Phi_{i,j}^{(t)} = \lfloor \mathbf{P}_{i,j}x_i^{(t)} - \mathbf{P}_{j,i}x_j^{(t)} \rfloor$ , provided that  $\mathbf{P}_{i,j}x_i^{(t)} \geq \mathbf{P}_{j,i}x_j^{(t)}$ . In other words, the flow on each edge is always rounded down. In our BED framework this would imply a  $\Lambda$  of order  $T$  after  $T$  time steps.

A simple *randomized rounding diffusion algorithm* chooses for  $\mathbf{P}_{i,j}x_i^{(t)} \geq \mathbf{P}_{j,i}x_j^{(t)}$  the flow  $\Phi_{i,j}^{(t)}$  as the randomized rounding of  $\mathbf{P}_{i,j}x_i^{(t)} - \mathbf{P}_{j,i}x_j^{(t)}$ , that is, it rounds up with probability  $(\mathbf{P}_{i,j}x_i^{(t)} - \mathbf{P}_{j,i}x_j^{(t)}) - \lfloor \mathbf{P}_{i,j}x_i^{(t)} - \mathbf{P}_{j,i}x_j^{(t)} \rfloor$  and rounds down otherwise. This typically achieves an error  $\Lambda$  of order  $\sqrt{T}$  after  $T$  time steps.

*Handling Negative Loads.* Unless there is a lower bound on the minimum load of a vertex, negative loads may occur during the balancing procedure. In the following, we describe a simple approach to cope with this problem.

Consider a graph  $G$  for which we can prove a deviation of at most  $\gamma$  from the idealized process. Let  $x^{(0)}$  be the initial load vector with an average load of  $\bar{x}$ . Then at the beginning of the balancing procedure, each node generates  $\gamma$  additional (virtual) tokens. During the balancing procedure, these tokens are regarded as common tokens,

but at the end they are ignored. First observe that since the minimum load at each node in the idealized process is at least  $\gamma$ , it follows that at each step, every node has at least a load of zero in the discrete process. Since each node has a load of  $\bar{x} + \mathcal{O}(\gamma)$  at the end, we end up with a load distribution where the maximum load is still  $\bar{x} + \mathcal{O}(\gamma)$  (ignoring the virtual tokens).

**3. Basic method to analyze our quasirandom algorithm.** To bound runtime and discrepancy of a BED algorithm, we always bound the deviation between the idealized model and the discrete model which is an important measure in its own right. For this, let  $x_\ell^{(t)}$  denote the load on vertex  $\ell$  in step  $t$  in the discrete model and  $\xi_\ell^{(t)}$  denote the load on vertex  $\ell$  in step  $t$  in the idealized model. We assume that the discrete and idealized model start with the same initial load, that is,  $x^{(0)} = \xi^{(0)}$ . As derived in Rabani et al. [32], their difference can be written as

$$x_\ell^{(t)} - \xi_\ell^{(t)} = \sum_{s=0}^{t-1} \sum_{[i:j] \in E} e_{i,j}^{(t-s)} (\mathbf{P}_{\ell,i}^s - \mathbf{P}_{\ell,j}^s). \quad (3.1)$$

where  $[i : j]$  refers to an edge  $\{i, j\} \in E$  with  $i < j$ , where “ $<$ ” is some arbitrary but fixed ordering on the vertices  $V$ . It will be sufficient to bound equation (3.1), as the convergence speed of the idealized process can be bounded in terms of the second largest eigenvalue.

**THEOREM 3.1** (e.g., [32, Thm. 1]). *On all graphs with second largest eigenvalue in absolute value  $\lambda_2 = \lambda_2(\mathbf{P})$ , the idealized process with divisible tokens reduces an initial discrepancy  $K$  to  $\ell$  within*

$$\frac{2}{1-\lambda_2} \ln \left( \frac{Kn^2}{\ell} \right)$$

*time steps.*

As  $\lambda_2 = 1 - \Theta(\log^{-1} n)$  for the hypercube and  $\lambda_2 = 1 - \Theta(n^{-2/d})$  for the  $d$ -dimensional torus [14], one immediately gets the following corollary.

**COROLLARY 3.2.** *The idealized process reduces an initial discrepancy of  $K$  to 1 within  $\mathcal{O}(n^{2/d} \log(Kn))$  time steps on the  $d$ -dimensional torus and within  $\mathcal{O}(\log n \log(Kn))$  time steps on the hypercube.*

An important property of all examined graph classes will be unimodality or log-concavity of certain transition probabilities.

**DEFINITION 3.3.** *A function  $f: \mathbb{N} \rightarrow \mathbb{R}_{\geq 0}$  is log-concave if  $f(i)^2 \geq f(i-1) \cdot f(i+1)$  for all  $i \in \mathbb{N}_{>0}$ .*

**DEFINITION 3.4.** *A function  $f: \mathbb{N} \rightarrow \mathbb{R}$  is unimodal if there is a  $s \in \mathbb{N}$  such that  $f|_{x \leq s}$  as well as  $f|_{x \geq s}$  are monotone.*

Log-concave functions are sometimes also called *strongly unimodal* [23]. We summarize some classical results regarding log-concavity and unimodality.

**FACT 3.5.**

- (i) *Let  $f$  be a log-concave function. Then,  $f$  is also a unimodal function (e.g. [22, 23]).*
- (ii) *Hoggar’s theorem [18]: Let  $f$  and  $g$  be log-concave functions. Then their convolution  $(f * g)(k) = \sum_{i=0}^k f(i) g(k-i)$  is also log-concave.*
- (iii) *Let  $f$  be a log-concave function and  $g$  be a unimodal function. Then their convolution  $f * g$  is a unimodal function [23].*

Our interest in unimodality is based on the fact that an alternating sum over a unimodal function can be bounded by their maximum. More precisely, for a non-negative and unimodal function  $f: X \rightarrow \mathbb{R}$  and  $t_0, \dots, t_k \in X$  with  $t_0 \leq \dots \leq t_k$ , the following

holds:

$$\left| \sum_{i=0}^k (-1)^i f(t_i) \right| \leq \max_{x \in X} f(x).$$

We generalize this well-known property in the following lemma.

LEMMA 3.6. *Let  $f: X \rightarrow \mathbb{R}$  be non-negative with  $X \subseteq \mathbb{R}$ . Let  $A_0, \dots, A_k \in \mathbb{R}$  and  $t_0, \dots, t_k \in X$  such that  $t_0 \leq \dots \leq t_k$  and  $|\sum_{i=a}^k A_i| \leq k$  for all  $0 \leq a \leq k$ . If  $f$  has  $\ell$  local extrema, then*

$$\left| \sum_{i=0}^k A_i f(t_i) \right| \leq (\ell + 1) k \max_{j=0}^k f(t_j).$$

*Proof.* Let us start with the assumption that  $f(t_i)$ ,  $0 \leq i \leq k$ , is monotone increasing. With  $f(t_{-1}) := 0$ , it is easy to see that then

$$\begin{aligned} \left| \sum_{i=0}^k A_i f(t_i) \right| &= \left| \sum_{i=0}^k \sum_{j=0}^i A_i (f(t_j) - f(t_{j-1})) \right| \\ &= \left| \sum_{j=0}^k \sum_{i=j}^k A_i (f(t_j) - f(t_{j-1})) \right| \\ &\leq \sum_{j=0}^k |f(t_j) - f(t_{j-1})| \left| \sum_{i=j}^k A_i \right| \\ &\leq \sum_{j=0}^k |f(t_j) - f(t_{j-1})| k \\ &= k \max_{j=0}^k f(t_j). \end{aligned}$$

The same holds if  $f(t_i)$ ,  $0 \leq i \leq k$ , is monotone decreasing. If  $f(x)$  has  $\ell$  local extrema, we split the sum in  $(\ell + 1)$  parts such that  $f(x)$  is monotone on each part and apply above arguments.  $\square$

*Random Walks.* To examine the diffusion process, it will be useful to define a random walk based on  $\mathbf{P}$ . For any pair of vertices  $i, j$ ,  $\mathbf{P}_{i,j}^t$  is the probability that a random walk guided by  $\mathbf{P}$  starting from  $i$  is located at  $j$  at step  $t$ . In the following Section 4 it will be useful to set  $\mathbf{P}_{i,j}(t) := \mathbf{P}_{i,j}^t$  and to denote with  $\mathbf{f}_{i,j}(t)$  for  $i \neq j$  the first-passage probabilities, that is, the probability that a random walk starting from  $i$  visits the vertex  $j$  at step  $t$  for the first time.

#### 4. Analysis on the hypercube.

We first give the definition of the hypercube.

DEFINITION 4.1. *A  $d$ -dimensional hypercube with  $n = 2^d$  vertices has vertex set  $V = \{0, 1\}^d$  and edge set  $E = \{\{i, j\} \mid i \text{ and } j \text{ differ in one bit}\}$ .*

In this section we prove the following result.

THEOREM 4.2. *For all initial load vectors on the  $d$ -dimensional hypercube with  $n$  vertices, the deviation between the idealized process and a discrete process with accumulated rounding errors at most  $\Lambda$  is upper bounded by  $4\Lambda \log n$  at all times and there are load vectors for which this deviation can be  $(\log n)/2$ .*

Recall that for BED algorithms  $\Lambda = \mathcal{O}(1)$ . With Theorem 3.1 it follows that any BED algorithm (and in particular our quasirandom algorithm) reduces the discrepancy of any initial load vector with discrepancy  $K$  to  $\mathcal{O}(\log n)$  within  $\mathcal{O}(\log n \log(Kn))$  time steps.

**4.1. Log-concave passage time on paths.** To prove Theorem 4.2, we first consider a discrete-time random walk on a path  $\mathcal{P} = (0, 1, \dots, d)$  starting at node 0. We make use of a special generating function, called *z-transform*. The *z-transform* of a function  $g: \mathbb{N} \mapsto \mathbb{R}_{\geq 0}$  is defined by  $\mathcal{G}(z) = \sum_{i=0}^{\infty} g(i) \cdot z^{-i}$ . We will use the fact that a convolution reduces to multiplication in the *z-plane*. Instead of the *z-transform* one could carry out a similar analysis using the *probability generating function*. We choose to use the *z-transform* here since it leads to slightly simpler arithmetic expressions.

Our analysis also uses the *geometric distribution* with parameter  $p$ , which is defined by  $\text{Geo}(p)(t) = (1-p)^{t-1}p$  for  $t \in \mathbb{N} \setminus \{0\}$  and  $\text{Geo}(p)(0) = 0$ . It is easy to check that  $\text{Geo}(p)$  is log-concave. Moreover, the *z-transform* of  $\text{Geo}(p)$  is

$$\sum_{i=1}^{\infty} \text{Geo}(p)(i) \cdot z^{-i} = \frac{p}{z - (1-p)}.$$

For each node  $i \in \mathcal{P}$ , let  $\alpha_i$  be the loop probability at node  $i$  and  $\beta_i$  be the *upward probability*, i.e., the probability to move to node  $i+1$ . Then, the *downward probability* at node  $i$  is  $1 - \alpha_i - \beta_i$ . We can assume that  $\beta_i > 0$  for all  $i \in \mathcal{P} \setminus \{d\}$ . We are interested in the first-passage probabilities  $\mathbf{f}_{0,d}(t)$ . Observe that

$$\mathbf{f}_{0,d}(t) = (\mathbf{f}_{0,1} * \mathbf{f}_{1,2} * \dots * \mathbf{f}_{d-1,d})(t). \quad (4.1)$$

In the following, we will show that  $\mathbf{f}_{0,d}(t)$  is *log-concave*. Indeed, we show a much stronger result:

**THEOREM 4.3.** *Consider a random walk on a path  $\mathcal{P} = (0, 1, \dots, d)$  starting at node 0. If  $\alpha_i \geq \frac{1}{2}$  for all nodes  $i \in \mathcal{P}$ , then  $\mathbf{f}_{0,d}$  can be expressed as convolution of  $d$  independent geometric distributions.*

As the geometric distribution is log-concave and the convolution of log-concave functions is again log-concave (cf. Fact 3.5), we immediately get the following corollary.

**COROLLARY 4.4.** *Consider a random walk on a path  $\mathcal{P} = (0, 1, \dots, d)$  starting at node 0. If  $\alpha_i \geq \frac{1}{2}$  for all nodes  $i \in \mathcal{P}$ , then  $\mathbf{f}_{0,d}(t)$  is log-concave in  $t$ .*

Note that Theorem 4.3 follows directly from Theorem 1.2 of Fill [10]. As Theorem 4.3 is a crucial ingredient for proving our main result Theorem 4.2 for the hypercube, we give a simpler alternative proof of the statement. While Fill's proof is purely stochastic, our proof is elementary and based on functional analysis of the *z-transform*. Our analysis for the discrete-time random walk should also be compared with Keilson's analysis of the continuous-time process [22]. The continuous-time process was independently considered even earlier by Karlin and McGregor [20].

Before proving the theorem, we will show how to obtain  $\mathbf{f}_{0,d}(t)$  by a recursive argument. For this, suppose we are at node  $i \in \mathcal{P} \setminus \{d\}$ . The next step is a loop with probability  $\alpha_i$ . Moreover, the next subsequent non-loop move ends at  $i+1$  with probability  $\frac{\beta_i}{1-\alpha_i}$  and at  $i-1$  with probability  $\frac{1-\beta_i-\alpha_i}{1-\alpha_i}$ . Thus, for all  $i \in \mathcal{P} \setminus \{d\}$ ,

$$\mathbf{f}_{i,i+1}(t) = \frac{\beta_i}{1-\alpha_i} \cdot \text{Geo}(1-\alpha_i)(t) + \frac{1-\beta_i-\alpha_i}{1-\alpha_i} \cdot (\text{Geo}(1-\alpha_i) * \mathbf{f}_{i-1,i} * \mathbf{f}_{i,i+1})(t),$$

with corresponding *z-transform*

$$\mathcal{F}_{i,i+1}(z) = \frac{\beta_i}{1-\alpha_i} \cdot \frac{1-\alpha_i}{z-\alpha_i} + \frac{1-\beta_i-\alpha_i}{1-\alpha_i} \cdot \frac{1-\alpha_i}{z-\alpha_i} \cdot \mathcal{F}_{i-1,i}(z) \cdot \mathcal{F}_{i,i+1}(z).$$

Rearranging terms yields

$$\mathcal{F}_{i,i+1}(z) = \frac{\beta_i}{z-\alpha_i - (1-\beta_i-\alpha_i) \cdot \mathcal{F}_{i-1,i}(z)}, \quad (4.2)$$

for all  $i \in \mathcal{P} \setminus \{d\}$ . So  $\mathcal{F}_{i,i+1}(z)$  is obtained recursively with  $\mathcal{F}_{0,1}(z) = \frac{\beta_0}{z-(1-\beta_0)}$ . Finally the  $z$ -transform of equation (4.1) is

$$\mathcal{F}_{0,d}(z) = \mathcal{F}_{0,1}(z) \cdot \mathcal{F}_{1,2}(z) \cdot \dots \cdot \mathcal{F}_{d-1,d}(z). \quad (4.3)$$

In the following, we prove some properties of  $\mathcal{F}_{i,i+1}(z)$  for  $i \in \mathcal{P} \setminus \{d\}$ .

LEMMA 4.5. *Except for singularities,  $\mathcal{F}_{i,i+1}(z)$  is monotone decreasing in  $z$ , for all  $i \in \mathcal{P} \setminus \{d\}$ .*

*Proof.* We will show the claim by induction on  $i$ . It is easy to see that the claim holds for the base case ( $i = 0$ ) since  $\mathcal{F}_{0,1}(z) = \frac{\beta_0}{z-(1-\beta_0)}$ . Assume inductively that the claim holds for  $\mathcal{F}_{i-1,i}(z)$ . With  $1 - \beta_i - \alpha_i \geq 0$  this directly implies that the denominator of equation (4.2) is increasing in  $z$ . The claim for  $\mathcal{F}_{i,i+1}(z)$  follows.  $\square$

LEMMA 4.6. *For all  $i \in \mathcal{P} \setminus \{d\}$ ,  $\mathcal{F}_{i,i+1}(z)$  has exactly  $i + 1$  poles which are all in the interval  $(0, 1)$ . The poles of  $\mathcal{F}_{i,i+1}(z)$  are distinct from the poles of  $\mathcal{F}_{i-1,i}(z)$ .*

*Proof.* Before proving the claims of the lemma, we will show that  $\mathcal{F}_{i,i+1}(0) \geq -1$  and  $\mathcal{F}_{i,i+1}(1) = 1$  for all  $i \in \mathcal{P} \setminus \{d\}$ . Observe, that  $\mathcal{F}_{0,1}(0) = \frac{\beta_0}{-(1-\beta_0)} = \frac{1-\alpha_0}{-\alpha_0} \geq -1$ , since  $\alpha_0 \geq \frac{1}{2}$ . Also observe that  $\mathcal{F}_{0,1}(1) = 1$ . Assume, inductively that  $\mathcal{F}_{i-1,i}(0) \geq -1$  and  $\mathcal{F}_{i-1,i}(1) = 1$ . Then with equation (4.2),  $\mathcal{F}_{i,i+1}(0) \geq \frac{\beta_i}{-\alpha_i-(1-\beta_i-\alpha_i)\cdot(-1)} = \frac{\beta_i}{1-2\alpha_i-\beta_i} \geq -1$ , since  $1 - 2\alpha_i \leq 0$ . Moreover,  $\mathcal{F}_{i,i+1}(1) = \frac{\beta_i}{1-\alpha_i-(1-\alpha_i-\beta_i)} = 1$ . Thus,  $\mathcal{F}_{i,i+1}(0) \geq -1$  and  $\mathcal{F}_{i,i+1}(1) = 1$  for all  $i \in \mathcal{P} \setminus \{d\}$ .

We will now show the claims of the lemma by induction. For the base case observe that  $\mathcal{F}_{0,1}(z) = \frac{\beta_0}{z-(1-\beta_0)}$  has one pole at  $z = 1 - \beta_0 > 0$  and  $\mathcal{F}_{-1,0}$  is not defined. This implies the claim for  $i = 0$ . Suppose the claim holds for  $\mathcal{F}_{i-1,i}(z)$  and let  $z_1, z_2, \dots, z_i$  be the poles of  $\mathcal{F}_{i-1,i}(z)$ . Without loss of generality, we assume  $0 < z_1 < z_2 < \dots < z_i < 1$ . Let  $g_i(z)$  be the denominator of equation (4.2), that is,

$$g_i(z) := z - \alpha_i - (1 - \beta_i - \alpha_i) \cdot \mathcal{F}_{i-1,i}(z).$$

Observe that

- (i)  $g_i(z)$  has the same set of poles as  $\mathcal{F}_{i-1,i}(z)$ ,
- (ii)  $\lim_{z \rightarrow -\infty} g_i(z) = -\infty$ , and
- (iii)  $\lim_{z \rightarrow \infty} g_i(z) = \infty$ .

By equation (4.2),  $\mathcal{F}_{i,i+1}(z)$  has its poles at the zeros of  $g_i(z)$ . Lemma 4.5 shows that in each interval  $(z_j, z_{j+1})$  with  $1 \leq j \leq i - 1$ ,  $g_i(z)$  is increasing in  $z$ . Using fact (i) this implies that  $g_i(z)$  has exactly one zero in each interval  $(z_j, z_{j+1})$ . Thus  $\mathcal{F}_{i,i+1}(z)$  has exactly one pole in each interval  $(z_j, z_{j+1})$ . Similarly, Lemma 4.5 with facts (i),(ii) and (iii) imply that  $\mathcal{F}_{i,i+1}(z)$  has exactly one pole, say  $z'$ , in the interval  $[-\infty, z_1)$  and one pole, say  $z''$  in the interval  $(z_i, \infty]$ . This implies that  $\mathcal{F}_{i,i+1}(z)$  has exactly  $i + 1$  poles which are all distinct from the poles of  $\mathcal{F}_{i-1,i}(z)$ . It remains to show that  $z' > 0$  and  $z'' < 1$ .

Since  $\mathcal{F}_{i-1,i}(0) \geq -1$  and  $\lim_{z \rightarrow -\infty} \mathcal{F}_{i-1,i}(z) = -0$  it follows with Lemma 4.5 that  $-1 \leq \mathcal{F}_{i-1,i}(z) \leq 0$  for all real  $z < 0$ . So  $g_i(z) < 0$  for all real  $z < 0$ . It follows that  $z' > 0$ . Similarly, since  $\mathcal{F}_{i-1,i}(1) = 1$  and  $\lim_{z \rightarrow \infty} \mathcal{F}_{i-1,i}(z) = +0$ , it follows with Lemma 4.5 that  $0 \leq \mathcal{F}_{i-1,i}(z) \leq 1$  for all real  $z > 1$ . So  $g_i(z) > 0$  for all real  $z > 1$ . It follows that  $z'' < 1$ . This finishes the proof of our inductive step. The claim follows.  $\square$

LEMMA 4.7. *Let  $(b_{j,i})_{j=0}^i$  be the poles of  $\mathcal{F}_{i,i+1}(z)$ ,  $i \in \mathcal{P} \setminus \{d\}$ , and define  $P_i(z) = \prod_{j=0}^i (z - b_{j,i})$ . Then  $\mathcal{F}_{i,i+1}(z) = \beta_i \cdot \frac{P_{i-1}(z)}{P_i(z)}$  for all  $i \in \mathcal{P} \setminus \{d\}$ .*



*Proof.* Our proof is by induction on  $i$ . For the base case ( $i = 0$ ), observe that  $P_{-1}(z) = 1$  and thus  $\mathcal{F}_{0,1}(z)$  has the desired form. Suppose the claim holds for  $\mathcal{F}_{i-1,i}(z)$ . Then equation (4.2) implies

$$\begin{aligned}\mathcal{F}_{i,i+1}(z) &= \frac{\beta_i}{z - \alpha_i - (1 - \beta_i - \alpha_i) \cdot \beta_{i-1} \cdot \frac{P_{i-2}(z)}{P_{i-1}(z)}} \\ &= \frac{\beta_i \cdot P_{i-1}(z)}{(z - \alpha_i) \cdot P_{i-1}(z) - (1 - \beta_i - \alpha_i) \cdot \beta_{i-1} \cdot P_{i-2}(z)}.\end{aligned}\quad (4.4)$$

Observe that  $(z - \alpha_i) \cdot P_{i-1}(z)$  is a polynomial of degree  $i + 1$  where the leading term has a coefficient of 1. This also holds for the denominator of equation (4.4), since there, we only subtract a polynomial of order  $i - 1$ . By Lemma 4.6 we know that  $\mathcal{F}_{i,i+1}(z)$  has exactly  $i + 1$  real positive poles. It follows that the denominator of equation (4.4) is equal to  $P_i(z)$ . The claim follows.  $\square$

We are now ready to prove Theorem 4.3.

*Proof of Theorem 4.3.* By equation (4.3) and Lemma 4.7, we get

$$\mathcal{F}_{0,d}(z) = \prod_{i=0}^{d-1} \mathcal{F}_{i,i+1}(z) = \prod_{i=0}^{d-1} \left( \beta_i \cdot \frac{P_{i-1}(z)}{P_i(z)} \right) = \frac{\prod_{i=0}^{d-1} \beta_i}{P_{d-1}(z)} = K_d \cdot \prod_{i=0}^{d-1} \frac{1 - b_{i,d-1}}{z - b_{i,d-1}},$$

where  $(b_{i,d-1})_{i=0}^{d-1}$  are the poles of  $\mathcal{F}_{d-1,d}(z)$  as defined in Lemma 4.7 and  $K_d = \prod_{i=0}^{d-1} \frac{\beta_i}{1 - b_{i,d-1}}$ . By Lemma 4.6,  $b_{i,d-1} \in (0, 1)$  for all  $i$ . Now for each  $i$  the term  $\frac{1 - b_{i,d-1}}{z - b_{i,d-1}}$  is the  $z$ -transform of the geometric distribution with parameter  $1 - b_{i,d-1}$ , i.e.,  $\text{Geo}(1 - b_{i,d-1})(t)$ .

Thus,  $\mathbf{f}_{0,d}(t)$  can be expressed as the convolution of  $d$  independent geometric distributions

$$\mathbf{f}_{0,d}(t) = K_d \cdot [\text{Geo}(1 - b_{0,d-1}) * \text{Geo}(1 - b_{1,d-1}) * \dots * \text{Geo}(1 - b_{d-1,d-1})](t).$$

Moreover, since  $\mathbf{f}_{0,d}$  is a probability distribution over  $t$  and the convolution of probability distributions is again a probability distribution, we have  $K_d = 1$ . The theorem follows.  $\square$

One should note that it follows from [10, Theorem 1.2] that the parameters  $(b_{i,d-1})_{i=0}^{d-1}$  in the geometric distributions are the eigenvalues of the underlying transition matrix.

Recall that our aim is to prove unimodality for the function  $\mathbf{P}_{0,j}^t$  (in  $t$ ). Using the simple convolution formula  $\mathbf{P}_{0,j} = \mathbf{f}_{0,j} * \mathbf{P}_{j,j}$  and the log-concavity of  $\mathbf{f}_{0,j}$ , it suffices to prove that  $\mathbf{P}_{j,j}$  is unimodal (cf. Fact 3.5). In the following, we will prove that  $\mathbf{P}_{j,j}$  is even non-increasing in  $t$ .

**LEMMA 4.8.** *Let  $\mathbf{P}$  be the  $(d + 1) \times (d + 1)$ -transition matrix defining an ergodic Markov chain on a path  $\mathcal{P} = (0, \dots, d)$ . If  $\mathbf{P}_{ii} \geq \frac{1}{2}$  for all  $0 \leq i \leq d$  then for all  $0 \leq i \leq d$ ,  $\mathbf{P}_{i,i}^t$  is non-increasing in  $t$ .*

*Proof.* It is well known that ergodic Markov chains on paths are time reversible (see e.g. Section 4.8 of Ross [33]). To see this let  $\pi = (\pi_0, \dots, \pi_d)$  be the stationary distribution. Then for all  $0 \leq i \leq d - 1$  the rate at which the process goes from  $i$  to  $i + 1$  (namely,  $\pi_i \mathbf{P}_{i,i+1}$ ) is equal to the rate at which the process goes from  $i + 1$  to  $i$  (namely,  $\pi_{i+1} \mathbf{P}_{i+1,i}$ ). Thus,  $\mathbf{P}$  is time-reversible.

One useful property of a time-reversible matrix is that its eigenvalues are all real. The Geršgorin disc theorem states that every eigenvalue  $\lambda_j$ ,  $0 \leq j \leq d$ , satisfies the condition

$$|\lambda_j - \mathbf{P}_{ii}| \leq 1 - \mathbf{P}_{ii},$$

for some  $0 \leq i \leq d$ . Since  $\mathbf{P}_{ii} \geq \frac{1}{2}$ , this directly implies that all eigenvalues are in the interval  $[0, 1]$ .

It is well-known that there is an orthonormal base of  $\mathbb{R}^{d+1}$  which is formed by the eigenvectors  $v_0, v_1, \dots, v_d$  (see e.g. [17]). Then for any  $n$ -dimensional vector  $w \in \mathbb{R}^{d+1}$ ,  $w = \sum_{j=0}^d \langle w, v_j \rangle v_j$ , where  $\langle \cdot, \cdot \rangle$  denotes the inner product. Applying this to the  $i$ -th unit vector  $e_i$  and using  $[\cdot]_i$  to denote the  $i$ -th entry of a vector in  $\mathbb{R}^{d+1}$  we obtain

$$e_i = \sum_{j=0}^d \langle e_i, v_j \rangle v_j = \sum_{j=0}^d [v_j]_i v_j.$$

Thus,

$$\mathbf{P}^t e_i = \mathbf{P}^t \left( \sum_{j=0}^d [v_j]_i v_j \right) = \sum_{j=0}^d [v_j]_i \mathbf{P}^t v_j = \sum_{j=0}^d [v_j]_i \lambda_j^t v_j$$

and finally

$$\mathbf{P}_{i,i}^t = [\mathbf{P}^t e_i]_i = \sum_{j=0}^d [v_j]_i \lambda_j^t [v_j]_i = \sum_{j=0}^d \lambda_j^t [v_j]_i^2,$$

which is non-increasing in  $t$  as  $[v_j]_i \in \mathbb{R}$  and  $0 \leq \lambda_j \leq 1$  for all  $0 \leq j \leq d$ .  $\square$

**4.2. Unimodal transition probabilities on the hypercube.** Combining Lemma 4.8 and Theorem 4.3 and then projecting the random walk on the hypercube on a random walk on a path, we obtain the following result.

**THEOREM 4.9.** *Let  $i, j \in V$  be two vertices of a  $d$ -dimensional hypercube. Then,  $\mathbf{P}_{i,j}(t)$  is unimodal.*

*Proof.* We use the following projection of a random walk on a  $d$ -dimensional hypercube with loop probability  $1/2$  to a random walk on a path with  $d$  vertices, again with loop probability  $1/2$ . The induced random walk is obtained from the mapping  $x \mapsto |x|_1$ , that is, vertices in  $\{0, 1\}^d$  with the same number of ones are equivalent. It is easy to check that this new random walk is a random walk on a path with vertices  $0, 1, \dots, d$  that moves right with probability  $\lambda_k = \frac{d-k}{2k}$ , left with probability  $\mu_k = \frac{d}{2k}$ , and loops with probability  $\frac{1}{2}$ . (This process is also known as Ehrenfest chain [16]).

Consider now the random walk on the path with vertex set  $\{0, 1, \dots, d\}$  and let  $j$  be an arbitrary number with  $0 \leq j \leq d$ . Recall that  $\mathbf{P}_{0,j}$  can be expressed as a convolution (cf. [16]) of  $\mathbf{P}$  and  $\mathbf{f}$  as follows,

$$\mathbf{P}_{0,j} = \mathbf{f}_{0,j} * \mathbf{P}_{j,j}.$$

By Corollary 4.4,  $\mathbf{f}_{0,j}(t)$  is log-concave. Moreover, Lemma 4.8 implies that  $\mathbf{P}_{j,j}(t)$  is non-increasing in  $t$  and hence unimodal. As the convolution of any log-concave function with any unimodal function is again unimodal (cf. Fact 3.5), it follows that  $\mathbf{P}_{0,j}(t)$  is unimodal in  $t$ .

Now fix two vertices  $i, j$  of the  $d$ -dimensional hypercube. By symmetry, we may assume that  $i = 0^d \equiv 0$ . Conditioned on the event that the projected random walk is

located at a vertex with  $|j|_1$  ones at step  $t$ , every vertex with  $|j|_1$  ones is equally likely. This gives  $\mathbf{P}_{0,j}(t) = \mathbf{P}_{0,|j|_1}(t) / \binom{d}{|j|_1}$ , and therefore the unimodality of  $\mathbf{P}_{0,|j|_1}(t)$  implies directly the unimodality of  $\mathbf{P}_{0,j}(t)$ , as needed.  $\square$

With more direct methods, one can prove the following supplementary result giving further insights into the distribution of  $\mathbf{P}_{i,j}(t)$ . As the result is not required for our analysis, the proof is given in the appendix.

**PROPOSITION 4.10.** *Let  $i, j \in V$  be two vertices of the  $d$ -dimensional hypercube with  $\text{dist}(i, j) \geq d/2$ . Then,  $\mathbf{P}_{i,j}(t)$  is monotone increasing.*

**4.3. Analysis of the discrete algorithm.** We are now ready to prove our main result for hypercubes.

*Proof of Theorem 4.2.* By symmetry, it suffices to bound the deviation at the vertex  $0 \equiv 0^d$ . Hence by equation (3.1) we have to bound

$$\begin{aligned} |x_0^{(t)} - \xi_0^{(t)}| &\leq \left| \sum_{s=0}^{t-1} \sum_{[i:j] \in E} e_{i,j}^{(t-s)} (\mathbf{P}_{0,i}(s) - \mathbf{P}_{0,j}(s)) \right| \\ &\leq \left| \sum_{s=0}^{t-1} \sum_{[i:j] \in E} e_{i,j}^{(t-s)} \mathbf{P}_{0,i}(s) \right| + \left| \sum_{s=0}^{t-1} \sum_{[i:j] \in E} e_{i,j}^{(t-s)} \mathbf{P}_{0,j}(s) \right| \\ &\leq \sum_{[i:j] \in E} \left| \sum_{s=0}^{t-1} e_{i,j}^{(t-s)} \mathbf{P}_{0,i}(s) \right| + \sum_{[i:j] \in E} \left| \sum_{s=0}^{t-1} e_{i,j}^{(t-s)} \mathbf{P}_{0,j}(s) \right|. \end{aligned}$$

Using Theorem 4.9, we know that the sequences  $\mathbf{P}_{0,i}(s)$  and  $\mathbf{P}_{0,j}(s)$  are unimodal in  $s$  and hence we can bound both summands by Lemma 3.6 (where  $\ell = 1$ ) to obtain that

$$\begin{aligned} |x_0^{(t)} - \xi_0^{(t)}| &\leq 2\Lambda \sum_{[i:j] \in E} \max_{s=0}^{t-1} \mathbf{P}_{0,i}(s) + 2\Lambda \sum_{[i:j] \in E} \max_{s=0}^{t-1} \mathbf{P}_{0,j}(s) \\ &= 2\Lambda d \sum_{i \in V} \max_{s=0}^{t-1} \mathbf{P}_{0,i}(s). \end{aligned} \quad (4.5)$$

To bound the last term, we view the random walk as the following process, similar to a balls-and-bins process. In each step  $t \in \mathbb{N}$  we choose a coordinate  $i \in \{1, \dots, d\}$  uniformly at random. Then with probability  $1/2$  we flip the bit of this coordinate; otherwise we keep it (equivalently, we set the bit to 1 with probability  $1/2$  and to zero otherwise).

Now we partition the random walk's distribution at step  $t$  according to the number of different coordinates chosen (not necessarily flipped) until step  $t$ . Consider  $\mathbf{P}_{0,x}(t)$  for a vertex  $x \in \{0, 1\}^d$ . Note that by the symmetry of the hypercube,  $\mathbf{P}_{0,x}(t)$  is the same for all  $x \in \{0, 1\}^d$  with the same  $|x|_1$ . Hence let us fix a value  $\ell$  with  $0 \leq \ell \leq d$  and let us consider  $\mathbf{P}_{0,\ell}(t)$  which is the probability for reaching the vertex, say,  $1^\ell 0^{d-\ell}$  from  $0 \equiv 0^d$  within  $t$  steps. Since (i) the  $k$  chosen coordinates must contain the  $\ell$  ones and (ii) all  $k$  chosen coordinates must be set to the correct value, we have

$$\mathbf{P}_{0,\ell}(t) = \sum_{k=\ell}^d \mathbf{Pr}[\text{exactly } k \text{ coordinates chosen in } t \text{ steps}] \cdot 2^{-k} \binom{d-\ell}{k-\ell} / \binom{d}{k}. \quad (4.6)$$

Using this to estimate  $\mathbf{P}_{0,i}(s)$ , we can bound equation (4.5) by

$$\begin{aligned} |x_0^{(t)} - \xi_0^{(t)}| &\leq 2\Lambda d \sum_{\ell=0}^d \binom{d}{\ell} \max_{s=0}^{\infty} \mathbf{P}_{0,\ell}(s) \\ &= 2\Lambda d \sum_{\ell=0}^d \binom{d}{\ell} \max_{s=0}^{\infty} \sum_{k=\ell}^d \mathbf{Pr}[\text{exactly } k \text{ coordinates chosen in } s \text{ steps}] \cdot \frac{\binom{d-\ell}{k-\ell}}{\binom{d}{k}} \cdot 2^{-k} \\ &\leq 2\Lambda d \sum_{\ell=0}^d \max_{k=\ell}^d \frac{\binom{d-\ell}{k-\ell} \binom{d}{\ell}}{\binom{d}{k}} 2^{-k}. \end{aligned}$$

The fraction in the last term corresponds to the probability of a hyper-geometric distribution and is therefore trivially upper-bounded by 1. This allows us to conclude that

$$|x_0^{(t)} - \xi_0^{(t)}| \leq 2\Lambda d \sum_{\ell=0}^d 2^{-\ell} \leq 4\Lambda d$$

and the first claim of the theorem follows.

The second claim follows by the following simple construction. Define a load vector  $x^{(0)}$  such that  $x_v^{(0)} := d$  for all vertices  $v = (v_1, v_2, \dots, v_d) \in \{0, 1\}^d$  with  $v_1 = 0$ , and  $x_v^{(0)} := 0$  otherwise. Then for each edge  $\{i, j\} \in E$  with  $0 = i_1 \neq j_1$  the fractional flow at step 1 is  $(\mathbf{P}_{i,j}x_i^{(0)} - \mathbf{P}_{i,j}x_j^{(0)}) = +\frac{1}{2}$ . Since in the first time step no rounding errors have been occurred so far, each edge is allowed to round up and down arbitrarily. Hence we can let all these edges round towards  $j$ , i.e.,  $\Phi_{i,j}^{(1)} := 1$  for each such edge  $\{i, j\} \in E$ . By definition, this implies for the corresponding rounding error,  $e_{i,j}^{(1)} = -\frac{1}{2}$ . Moreover, we have the following load distribution after step 1. We have  $x_v^{(1)} = 0$  for all vertices  $v$  if  $v_1 = 0$ , and  $x_v^{(1)} = d$  otherwise. Similarly, the fractional flow for each edge  $\{i, j\} \in E$  with  $0 = i_1 \neq j_1$  is  $(\mathbf{P}_{i,j}x_i^{(0)} - \mathbf{P}_{i,j}x_j^{(0)}) = -\frac{1}{2}$ . Since  $e_{i,j}^{(1)} = -\frac{1}{2}$ ,  $|\sum_{s=1}^2 e_{i,j}^{(s)}|$  will be minimized if  $e_{i,j}^{(2)} = \frac{1}{2}$ . Hence we can set  $\Phi_{i,j}^{(2)} := -1$ . This implies that we end up in exactly the same situation as at the beginning: the load vector is the same and also the sum over the previous rounding errors along each edge is zero. We conclude that there is an instance of the quasirandom algorithm for which  $x^{(t)} = x^{(t \bmod 2)}$ . This gives the claim.  $\square$

**5. Analysis on  $d$ -dimensional torus graphs.** We start this section with the formal definition of a  $d$ -dimensional torus.

**DEFINITION 5.1.** *A  $d$ -dimensional torus  $T(n_1, n_2, \dots, n_d)$  with  $n = n_1 \cdot n_2 \cdot \dots \cdot n_d$  vertices has vertex set  $V = \{0, 1, \dots, n_1 - 1\} \times \{0, 1, \dots, n_2 - 1\} \times \dots \times \{0, 1, \dots, n_d - 1\}$  and every vertex  $(i_1, i_2, \dots, i_d) \in V$  has  $2d$  neighbors  $((i_1 \pm 1) \bmod n_1, i_2, \dots, i_d)$ ,  $(i_1, (i_2 \pm 1) \bmod n_2, i_3, \dots, i_d)$ ,  $\dots$ ,  $(i_1, i_2, \dots, i_{d-1}, (i_d \pm 1) \bmod n_d)$ . Henceforth, we will always assume that  $d = \mathcal{O}(1)$ . We call a torus uniform if  $n_1 = n_2 = \dots = n_d = \sqrt[d]{n}$ .*

Without loss of generality we will assume in the remainder that  $n_1 \leq n_2 \leq \dots \leq n_d$ . By the symmetry of the torus this does not restrict our results.

Recall that  $\lambda_2$  denotes the second largest eigenvalue in absolute value. Before we analyze the deviation between the idealized and discrete process, we estimate  $(1 - \lambda_2)^{-1}$  for general torus graphs.

**LEMMA 5.2.** *For a  $d$ -dimensional torus  $T = T(n_1, n_2, \dots, n_d)$ ,  $(1 - \lambda_2)^{-1} = \Theta(n_d^2)$ .*

*Proof.* Following the notation of [3], for a  $k$ -regular graph  $G$ , let  $\mathbf{L}(G)$  be the matrix given by  $\mathbf{L}_{u,u}(G) = 1$ ,  $\mathbf{L}_{u,v}(G) = -\frac{1}{k}$  if  $\{u, v\} \in E(G)$  and  $\mathbf{L}_{u,v}(G) = 0$  otherwise. Let  $C_q$  be a cycle with  $q$  vertices. As shown in [3, Example 1.4], the eigenvalues of  $\mathbf{L}(C_q)$  are  $1 - \cos\left(\frac{2\pi r}{q}\right)$  where  $0 \leq r \leq q - 1$ . In particular, the second smallest eigenvalue of  $\mathbf{L}(C_q)$  denoted by  $\tau$  is given by  $1 - \cos\left(\frac{2\pi}{q}\right)$ .

Let  $\times$  denote the Cartesian product of graphs, that is, for any two graphs  $G_1 = (V_1, E_1)$ ,  $G_2 = (V_2, E_2)$  the graph  $G := G_1 \times G_2$  with  $G = (V, E)$  is defined by  $V = V_1 \times V_2$  and

$$E := \{((u_1, u_2), (v_1, u_2)) : u_2 \in V_2 \wedge \{u_1, v_1\} \in E_1\} \cup \{((u_1, u_2), (u_1, v_2)) : u_1 \in V_1 \wedge \{u_2, v_2\} \in E_2\}.$$

It is straightforward to generalize this definition to the Cartesian product of more than two graphs and it is then easy to check that  $T(n_1, n_2, \dots, n_d) = C_{n_1} \times C_{n_2} \times \dots \times C_{n_d}$ . The following theorem expresses the second smallest eigenvalue of the Cartesian product of graphs in terms of the second smallest eigenvalue of the respective graphs.

**THEOREM 5.3** ([3, Theorem 2.12]). *Let  $G_1, G_2, \dots, G_d$  be  $d$  graphs and let  $\tau_1, \tau_2, \dots, \tau_d$  be the respective second smallest eigenvalue of  $\mathbf{L}(G_1), \mathbf{L}(G_2), \dots, \mathbf{L}(G_d)$ . Then the second smallest eigenvalue  $\tau$  of  $\mathbf{L}(G_1 \times G_2 \times \dots \times G_d)$  satisfies  $\tau = \frac{1}{d} \min_{k=1}^d \tau_k$ .*

Applying this theorem to our setting, it follows that the second smallest eigenvalue  $\tau$  of  $\mathbf{L}(T)$  is  $\tau = \frac{1}{d} (1 - \cos(\frac{2\pi}{n_d}))$ . As  $n_d \geq \sqrt[d]{n}$ , we have  $\cos(\frac{2\pi}{n_d}) = 1 - \Theta(\frac{1}{n_d^2})$ . Using this and the fact that  $d$  is a constant, we obtain that  $\tau = \Theta(\frac{1}{n_d^2})$ . As  $T$  is a  $k$ -regular graph, the transition matrix  $\mathbf{P}(T)$  can be expressed as  $\mathbf{P}(T) = \mathbf{I} - \frac{1}{2}\mathbf{L}(T)$ . This implies for the second smallest eigenvalue of  $\mathbf{L}(T)$ ,  $\tau$ , and the second largest eigenvalue of the transition matrix  $\mathbf{P}(T)$ ,  $\lambda_2$ , that  $\lambda_2 = 1 - \frac{1}{2}\tau$ . Hence  $\lambda_2 = 1 - \Theta(\frac{1}{n_d^2})$ , which completes the proof.  $\square$

Note that the corresponding results of [11, 32] only hold for uniform torus graphs while the following result for our algorithm holds for general torus graphs.

**THEOREM 5.4.** *For all initial load vectors on the (not necessarily uniform)  $d$ -dimensional torus graph with  $n$  vertices, the deviation between the idealized process and a discrete process with accumulated rounding error at most  $\Lambda$  is  $\mathcal{O}(\Lambda)$  at all times.*

For any torus graph, we know that  $(1 - \lambda_2)^{-1} = \Theta(n_d^2)$  by Lemma 5.2. With Theorem 3.1 it follows that any BED algorithm (and in particular our quasirandom algorithm) reduces the discrepancy of any initial load vector with discrepancy  $K$  to  $\mathcal{O}(1)$  within  $\mathcal{O}(n_d^2 \log(Kn))$  time steps (for uniform torus graphs, this number of time steps is  $\mathcal{O}(n^{2/d} \log(Kn))$ ).

*Proof of Theorem 5.4.* By symmetry of the torus graph, we have  $\mathbf{P}_{i,j} = \mathbf{P}_{0,i-j}$ . Hence we set  $\mathbf{P}_i = \mathbf{P}_{0,i}$ . We will first reduce the random walk  $\mathbf{P}_{i,j}$  on the finite  $d$ -dimensional torus to a random walk on the infinite grid  $\mathbb{Z}^d$ , both with loop probability  $1/2$ . Let  $\bar{\mathbf{P}}_{i,j}$  be the transition probability from  $i$  to  $j$  on  $\mathbb{Z}^d$  defined by  $\bar{\mathbf{P}}_{i,j} = 1/(4d)$  if  $|i - j|_1 = 1$ ,  $\bar{\mathbf{P}}_{i,i} = 1/2$ , and 0 otherwise. For  $i = (i_1, \dots, i_d) \in V$  we set

$$H(i) := (i_1 + n_1 \mathbb{Z}, i_2 + n_2 \mathbb{Z}, \dots, i_d + n_d \mathbb{Z}) \subset \mathbb{Z}^d.$$

With  $\bar{\mathbf{P}}_i := \bar{\mathbf{P}}_{0,i}$ , we observe

$$\mathbf{P}_i^s = \sum_{k \in H(i)} \bar{\mathbf{P}}_k^s$$

for all  $s \geq 0$  and  $i \in V$ . We extend the definition of  $e_{i,j}$  in the natural way by setting

$$e_{k,\ell} := e_{i,j} \text{ for all } i, j \in V \text{ and } k \in H(i), \ell \in H(j).$$

Let  $\text{ARR} = \{\pm u_\ell \mid \ell \in \{1, \dots, d\}\} \in \mathbb{Z}^d$  with  $u_\ell$  being the  $\ell$ -th unit vector. Following equation (3.1) and using the fact that by symmetry it suffices to bound the deviation at the vertex  $0 := 0^d$ , we get

$$\begin{aligned} x_0^{(t)} - \xi_0^{(t)} &= \frac{1}{2} \sum_{s=0}^{t-1} \sum_{i \in V} \sum_{z \in \text{ARR}} e_{i,i+z}^{(t-s)} (\mathbf{P}_i^s - \mathbf{P}_{i+z}^s) \\ &= \frac{1}{2} \sum_{s=0}^{t-1} \sum_{i \in V} \sum_{z \in \text{ARR}} e_{i,i+z}^{(t-s)} \left( \sum_{k \in H(i)} \bar{\mathbf{P}}_k^s - \sum_{\ell \in H(i+z)} \bar{\mathbf{P}}_\ell^s \right) \\ &= \frac{1}{2} \sum_{s=0}^{t-1} \sum_{z \in \text{ARR}} \sum_{i \in V} e_{i,i+z}^{(t-s)} \left( \sum_{k \in H(i)} \bar{\mathbf{P}}_k^s - \bar{\mathbf{P}}_{k+z}^s \right) \\ &= \frac{1}{2} \sum_{i \in V} \sum_{z \in \text{ARR}} \sum_{k \in H(i)} \sum_{s=0}^{t-1} e_{k,k+z}^{(t-s)} (\bar{\mathbf{P}}_k^s - \bar{\mathbf{P}}_{k+z}^s) \end{aligned}$$

As  $\mathbb{Z}^d = \bigcup_{i \in V} H(i)$  is a disjoint union, we can also write

$$x_0^{(t)} - \xi_0^{(t)} = \frac{1}{2} \sum_{k \in \mathbb{Z}^d} \sum_{z \in \text{ARR}} \sum_{s=0}^{t-1} e_{k, k+z}^{(t-s)} (\bar{\mathbf{P}}_k^s - \bar{\mathbf{P}}_{k+z}^s). \quad (5.1)$$

We now carefully break down the sums of equation (5.1) and show that each part can be bounded by  $\mathcal{O}(\Lambda)$ . For this, our main tool will be Lemma 3.6. As we cannot prove unimodality of  $(\bar{\mathbf{P}}_k^s - \bar{\mathbf{P}}_{k+z}^s)$  directly, we will use an appropriate local central limit theorems to approximate the transition probabilities  $\bar{\mathbf{P}}_k^s$  of  $\mathbb{Z}^d$  with a multivariate normal distribution. To derive the limiting distribution  $\bar{\mathbf{P}}_k^s$  of our random walk  $\bar{\mathbf{P}}_{i,j}$ , we follow Lawler and Limic [25] and let  $X = (X_1, \dots, X_d)$  be a  $\mathbb{Z}^d$ -valued random variable with  $\mathbf{Pr}[X = z] = 1/(4d)$  for all  $z \in \text{ARR}$  and  $\mathbf{Pr}[X = 0^d] = 1/2$ . Observe that  $\mathbf{E}[X_j X_k] = 0$  for  $j \neq k$  since not both of them can be non-zero simultaneously. Moreover,  $\mathbf{E}[X_j X_j] = \frac{1}{4d}(-1)^2 + \frac{1}{4d}(+1)^2 = \frac{1}{2d}$  for all  $1 \leq j \leq d$ . Hence the covariance matrix is

$$\Gamma := \left[ \mathbf{E}[X_j X_k] \right]_{1 \leq j, k \leq d} = (2d)^{-1} I.$$

From Eq. (2.2) of Lawler and Limic [25] we get

$$\tilde{\mathbf{P}}_k^s = \frac{1}{(2\pi)^d s^{d/2}} \int_{\mathbb{R}^d} \exp\left(\mathbf{i} \frac{x \cdot k}{\sqrt{s}}\right) \exp\left(-\frac{x \cdot \Gamma x}{2}\right) d^d x$$

where  $\mathbf{i} = \sqrt{-1}$  denotes the imaginary unit. With this we can further conclude that

$$\begin{aligned} \tilde{\mathbf{P}}_k^s &= \frac{1}{(2\pi)^d s^{d/2}} \int_{\mathbb{R}^d} \exp\left(\mathbf{i} \frac{x \cdot k}{\sqrt{s}} - \frac{x \cdot \Gamma x}{2}\right) d^d x \\ &= \frac{1}{(2\pi)^d s^{d/2}} \int_{\mathbb{R}^d} \exp\left(\mathbf{i} \frac{x \cdot k}{\sqrt{s}} - \frac{\|x\|_2^2}{4d}\right) d^d x \\ &= \frac{1}{(2\pi)^d s^{d/2}} \int_{\mathbb{R}^d} \exp\left(-\frac{1}{4d} \left(\|x\|_2^2 - 2\mathbf{i} \frac{2d}{\sqrt{s}} x \cdot k\right)\right) d^d x. \end{aligned} \quad (5.2)$$

To evaluate the integral we complete the square, which yields

$$\begin{aligned} &\int_{\mathbb{R}^d} \exp\left(-\frac{1}{4d} \left(\|x\|_2^2 - 2\mathbf{i} \frac{2d}{\sqrt{s}} x \cdot k\right)\right) d^d x \\ &= \int_{\mathbb{R}^d} \exp\left(-\frac{1}{4d} \left(\|x\|_2^2 - 2\mathbf{i} \frac{2d}{\sqrt{s}} x \cdot k - \frac{4d^2}{s} \|k\|_2^2 + \frac{4d^2}{s} \|k\|_2^2\right)\right) d^d x \\ &= \exp\left(-\frac{d}{s} \|k\|_2^2\right) \int_{\mathbb{R}^d} \exp\left(-\frac{1}{4d} \left\|x - \mathbf{i} \frac{2d}{\sqrt{s}} k\right\|_2^2\right) d^d x. \end{aligned} \quad (5.3)$$

By substituting  $z = x - \mathbf{i} \frac{2d}{\sqrt{s}} k$  we get

$$\begin{aligned} &\int_{\mathbb{R}^d} \exp\left(-\frac{1}{4d} \left\|x - \mathbf{i} \frac{2d}{\sqrt{s}} k\right\|_2^2\right) d^d x \\ &= \int_{\mathbb{R}^d} \exp\left(-\frac{1}{4d} (\|z\|_2^2)\right) d^d z \\ &= \int \cdots \int_{\mathbb{R}^d} \exp\left(-\frac{1}{4d} \left(\sum_{i=1}^d z_i^2\right)\right) dz_d \cdots dz_1 \end{aligned}$$

$$\begin{aligned}
&= \int \cdots \int_{\mathbb{R}^{d-1}} \exp \left( -\frac{1}{4d} \left( \sum_{i=1}^{d-1} z_i^2 \right) \right) \left( \int_{\mathbb{R}} \exp \left( -\frac{1}{4d} z_d^2 \right) dz_d \right) dz_{d-1} \dots dz_1 \\
&= (2\sqrt{\pi d}) \cdot \int \cdots \int_{\mathbb{R}^{d-1}} \exp \left( -\frac{1}{4d} \left( \sum_{i=1}^{d-1} z_i^2 \right) \right) dz_{d-1} \dots dz_1 \\
&= (2\sqrt{\pi d})^d.
\end{aligned} \tag{5.4}$$

Combining equations (5.2), (5.3), and (5.4), we get

$$\begin{aligned}
\tilde{\mathbf{P}}_k^s &= \frac{1}{(2\pi)^d s^{d/2}} \exp \left( -\frac{d}{s} \|k\|_2^2 \right) (2\sqrt{\pi d})^d \\
&= \left( \frac{d}{\pi s} \right)^{d/2} \exp \left( \frac{-d \|k\|_2^2}{s} \right).
\end{aligned} \tag{5.5}$$

It follows directly from Claims 4 and 5 of Cooper and Spencer [4] that for all  $k \in \mathbb{Z}^d$ ,  $z \in \text{ARR}$ ,

$$\tilde{\mathbf{P}}_k^s - \tilde{\mathbf{P}}_{k+z}^s = \mathcal{O}(\|k\|_2^{-(d+1)}) \text{ for all } s, \tag{5.6}$$

$$(s \mapsto \tilde{\mathbf{P}}_k^s - \tilde{\mathbf{P}}_{k+z}^s) \text{ has only a constant number of local extrema.} \tag{5.7}$$

This gives the intuition that by approximating  $(\bar{\mathbf{P}}_k^s - \bar{\mathbf{P}}_{k+z}^s)$  with  $(\tilde{\mathbf{P}}_k^s - \tilde{\mathbf{P}}_{k+z}^s)$ , we can bound equation (5.1) for sufficiently large  $k$  and  $s$  by Lemma 3.6. This approximation is made precise by the following local central limit theorems. Theorem 2.3.6 of Lawler and Limic [25] gives for all  $k \in \mathbb{Z}^d$ ,  $z \in \text{ARR}$ ,  $s \geq 0$ ,

$$|(\bar{\mathbf{P}}_k^s - \bar{\mathbf{P}}_{k+z}^s) - (\tilde{\mathbf{P}}_k^s - \tilde{\mathbf{P}}_{k+z}^s)| = \mathcal{O}(s^{-(d+3)/2}). \tag{5.8}$$

We first separate the case  $k = 0$  in equation (5.1). With  $\mathbb{Z}_{\neq 0}^d := \mathbb{Z}^d \setminus \{0^d\}$

$$\begin{aligned}
&\overset{(5.9a)}{\left| x_0^{(t)} - \xi_0^{(t)} \right|} \leq \frac{1}{2} \left| \sum_{z \in \text{ARR}} \sum_{s=0}^{t-1} e_{0,0+z}^{(t-s)} (\bar{\mathbf{P}}_0^s - \bar{\mathbf{P}}_{0+z}^s) \right| \\
&\quad + \underbrace{\frac{1}{2} \left| \sum_{k \in \mathbb{Z}_{\neq 0}^d} \sum_{z \in \text{ARR}} \sum_{s=0}^{t-1} e_{k,k+z}^{(t-s)} (\bar{\mathbf{P}}_k^s - \bar{\mathbf{P}}_{k+z}^s) \right|}_{(5.9b)} \tag{5.9}
\end{aligned}$$

Now we can apply the local central limit theorem given in equation (5.8) to (5.9a) and get

$$(5.9a) = \left| \sum_{z \in \text{ARR}} \sum_{s=0}^{t-1} e_{0,0+z}^{(t-s)} (\tilde{\mathbf{P}}_0^s - \tilde{\mathbf{P}}_{0+z}^s) \right| + \left| \sum_{z \in \text{ARR}} \sum_{s=0}^{t-1} \mathcal{O}(s^{-(d+3)/2}) \right| = \mathcal{O}(\Lambda),$$

where the last equality follows by Lemma 3.6 combined with equation (5.7) and the property  $|\sum_{s=1}^t e_{i,j}^{(s)}| \leq \Lambda$ .

We proceed by fixing a cutoff point  $T(k) := \frac{C \|k\|_2^2}{\ln^2(\|k\|_2)}$ ,  $k \in \mathbb{Z}_{\neq 0}^d$ , of the innermost sum of (5.9b) for some sufficiently small constant  $C > 0$ ,

$$\begin{aligned}
(5.9b) &\leq \underbrace{\left[ \sum_{k \in \mathbb{Z}_{\neq 0}^d} \sum_{z \in \text{ARR}} \sum_{s=1}^{T(k)} e_{k,k+z}^{(t-s)} \left( \bar{\mathbf{P}}_k^s - \bar{\mathbf{P}}_{k+z}^s \right) \right]}_{(5.10a)} \\
&\quad + \underbrace{\left[ \sum_{k \in \mathbb{Z}_{\neq 0}^d} \sum_{z \in \text{ARR}} \sum_{s=T(k)}^{t-1} e_{k,k+z}^{(t-s)} \left( \bar{\mathbf{P}}_k^s - \bar{\mathbf{P}}_{k+z}^s \right) \right]}_{(5.10b)}. \tag{5.10}
\end{aligned}$$

Note that the summand with  $s = 0$  is zero and can be ignored. The first summand (5.10a) can be bounded by

$$(5.10a) = \mathcal{O} \left( \sum_{k \in \mathbb{Z}_{\neq 0}^d} \sum_{z \in \text{ARR}} \sum_{s=1}^{T(k)} \left( \bar{\mathbf{P}}_k^s + \bar{\mathbf{P}}_{k+z}^s \right) \right). \tag{5.11}$$

It is known from Lawler [24, Lem. 1.5.1(a)] that for random walks on infinite grids,  $\sum_{\|k\|_2 \geq \lambda \sqrt{s}} \bar{\mathbf{P}}_k^s = \mathcal{O}(e^{-\lambda})$  for all  $s > 0$  and  $\lambda > 0$ . Hence also

$$\bar{\mathbf{P}}_k^s = \mathcal{O}(\exp(-\|k\|_2/\sqrt{s})) \text{ for all } s > 0, k \in \mathbb{Z}^d.$$

With this we can now bound the term  $(\bar{\mathbf{P}}_k^s + \bar{\mathbf{P}}_{k+z}^s)$  from equation (5.11). For  $0 < s \leq T(k)$ ,  $k \in \mathbb{Z}_{\neq 0}^d$ ,  $z \in \text{ARR}$ , and sufficiently small  $C > 0$ ,

$$\begin{aligned}
\bar{\mathbf{P}}_k^s + \bar{\mathbf{P}}_{k+z}^s &= \mathcal{O} \left( \exp \left( -\frac{\|k\|_2}{\sqrt{s}} \right) + \exp \left( -\frac{\|k-z\|_2}{\sqrt{s}} \right) \right) \\
&= \mathcal{O} \left( \exp \left( -\frac{\|k\|_2 - \|z\|_2}{\sqrt{s}} \right) \right) \\
&= \mathcal{O} \left( \exp \left( -\ln(\|k\|_2) \frac{(\|k\|_2 - 1)}{\sqrt{C} \|k\|_2} \right) \right) \\
&= \mathcal{O}(\|k\|_2^{-(d+4)}).
\end{aligned}$$

Plugging this into equation (5.11), we obtain that

$$(5.10a) = \mathcal{O} \left( \sum_{k \in \mathbb{Z}_{\neq 0}^d} T(k) \|k\|_2^{-(d+4)} \right) = \mathcal{O} \left( \sum_{k \in \mathbb{Z}_{\neq 0}^d} \|k\|_2^{-(d+2)} \ln^{-2}(\|k\|_2) \right) = \mathcal{O}(1).$$

To bound (5.10b), we approximate the transition probabilities of  $\mathbb{Z}^d$  with the multivariate normal distribution of equation (5.5) by the local central limit theorem stated



in equation (5.8),

$$\begin{aligned}
(5.10b) &= \left| \sum_{k \in \mathbb{Z}_{\neq 0}^d} \sum_{z \in \text{ARR}} \sum_{s=T(k)}^{t-1} e_{k,k+z}^{(t-s)} \left( \tilde{\mathbf{P}}_k^s - \tilde{\mathbf{P}}_{k+z}^s \right) \right. \\
&\quad \left. + \sum_{k \in \mathbb{Z}_{\neq 0}^d} \sum_{z \in \text{ARR}} \sum_{s=T(k)}^{t-1} e_{k,k+z}^{(t-s)} \left( \bar{\mathbf{P}}_k^s - \bar{\mathbf{P}}_{k+z}^s \right) - \left( \tilde{\mathbf{P}}_k^s - \tilde{\mathbf{P}}_{k+z}^s \right) \right| \\
&\quad \quad \quad (5.12a) \\
&\leq \underbrace{\left| \sum_{k \in \mathbb{Z}_{\neq 0}^d} \sum_{z \in \text{ARR}} \sum_{s=T(k)}^{t-1} e_{k,k+z}^{(t-s)} \left( \tilde{\mathbf{P}}_k^s - \tilde{\mathbf{P}}_{k+z}^s \right) \right|}_{(5.12b)} \\
&\quad + \underbrace{\left| \sum_{k \in \mathbb{Z}_{\neq 0}^d} \sum_{z \in \text{ARR}} \sum_{s=T(k)}^{t-1} e_{k,k+z}^{(t-s)} \mathcal{O}(s^{-(d+3)/2}) \right|}_{(5.12b)}. \tag{5.12}
\end{aligned}$$

We can bound the second term (5.12b) by

$$\begin{aligned}
(5.12b) &= \mathcal{O} \left( d \sum_{k \in \mathbb{Z}_{\neq 0}^d} \sum_{s=T(k)}^{\infty} s^{-(d+3)/2} \right) \\
&= \mathcal{O} \left( \sum_{k \in \mathbb{Z}_{\neq 0}^d} T(k)^{-(d+1)/2} \right) \\
&= \mathcal{O} \left( \sum_{k \in \mathbb{Z}_{\neq 0}^d} \frac{\ln^{d-1}(\|k\|_2)}{\|k\|_2^{d+1}} \right).
\end{aligned}$$

As there are constants  $C' > 0$  and  $\varepsilon > 0$  such that  $\ln^{d-1}(\|k\|_2) \leq C' \|k\|_2^{1-\varepsilon}$  for all  $k \in \mathbb{Z}_{\neq 0}^d$  we obtain that

$$(5.12b) = \mathcal{O} \left( \sum_{k \in \mathbb{Z}_{\neq 0}^d} \|k\|_2^{-(d+\varepsilon)} \right).$$

To see that this can be bounded by  $\mathcal{O}(1)$ , observe that with  $\mathbb{N}_{\neq 0}^d := \mathbb{N}^d \setminus \{0^d\}$ ,

$$\sum_{k \in \mathbb{Z}_{\neq 0}^d} \|k\|_2^{-(d+\varepsilon)} \leq 2^d \sum_{k \in \mathbb{N}_{\neq 0}^d} (k_1^2 + \dots + k_d^2)^{-(d+\varepsilon)/2}.$$

By convexity of  $x \mapsto x^2$ ,  $k_1^2 + \dots + k_d^2 \geq \frac{1}{d}(k_1 + \dots + k_d)^2$ , we then get

$$\begin{aligned}
(5.12b) &= \mathcal{O} \left( \sum_{k \in \mathbb{N}_{\neq 0}^d} (k_1 + \dots + k_d)^{-(d+\varepsilon)} \right) = \mathcal{O} \left( \sum_{x=1}^{\infty} \sum_{\substack{k \in \mathbb{N}^d \\ \|k\|_1 = x}} x^{-(d+\varepsilon)} \right) \\
&= \mathcal{O} \left( \sum_{x=1}^{\infty} x^{d-1} \cdot x^{-(d+\varepsilon)} \right) = \mathcal{O} \left( \sum_{x=1}^{\infty} x^{-(1+\varepsilon)} \right) = \mathcal{O}(1).
\end{aligned}$$

To finally bound (5.12a), we apply equation (5.7). We also use that  $\tilde{\mathbf{P}}_k^s - \tilde{\mathbf{P}}_{k+z}^s$  can be bounded by  $\mathcal{O}(\|k\|_2^{-(d+1)})$  according to equation (5.6). As  $|\sum_{s=1}^t e_{i,j}^{(s)}| \leq \Lambda$ , applying Lemma 3.6 yields

$$\begin{aligned}
(5.12a) &= \mathcal{O}\left(\sum_{k \in \mathbb{Z}_{\neq 0}^d} \sum_{z \in \text{ARR}} \Lambda \max_{s=T(k)}^{t-1} \left(\tilde{\mathbf{P}}_k^s - \tilde{\mathbf{P}}_{k+z}^s\right)\right) \\
&= \mathcal{O}\left(\sum_{k \in \mathbb{Z}_{\neq 0}^d} \sum_{z \in \text{ARR}} \Lambda \|k\|_2^{-(d+1)}\right) \\
&= \mathcal{O}\left(\Lambda d \sum_{k \in \mathbb{Z}_{\neq 0}^d} \|k\|_2^{-(d+1)}\right) \\
&= \mathcal{O}(\Lambda).
\end{aligned}$$

Combining all above bounds, we can conclude that  $|x_0^{(t)} - \xi_0^{(t)}| = \mathcal{O}(\Lambda)$ , meaning that the deviation between the idealized process and the discrete process at any time and vertex is at most  $\mathcal{O}(\Lambda)$ .  $\square$

**6. Lower bounds for previous algorithms.** For a better comparison with previous algorithms, this section gives lower bounds for other discrete diffusion processes. First, we observe the following general lower bound on the discrepancy for the RSW-algorithm.

**PROPOSITION 6.1.** *On all graphs  $G$  with maximum degree  $\Delta$ , there is an initial load-vector  $x^{(0)}$  with discrepancy  $\Delta \text{diam}(G)$  such that for the RSW-algorithm,  $x^{(t)} = x^{(t-1)}$  for all  $t \in \mathbb{N}$ .*

*Proof.* Fix a pair of vertices  $i$  and  $j$  with  $\text{dist}(i, j) = \text{diam}(G)$ . Define an initial load-vector  $x^{(0)}$  by

$$x_k^{(0)} := \text{dist}(k, i) \cdot \Delta.$$

Clearly, the initial discrepancy is  $x_j^{(0)} - x_i^{(0)} = \Delta \text{diam}(G)$ . We claim that  $x^{(1)} = x^{(0)}$ . Consider an arbitrary edge  $\{r, s\} \in E(G)$ . Then,

$$|\mathbf{P}_{r,s} x_r^{(1)} - \mathbf{P}_{s,r} x_s^{(1)}| = \frac{1}{2\Delta} |x_r^{(0)} - x_s^{(0)}| \leq \frac{1}{2\Delta} \Delta = \frac{1}{2}.$$

Hence the integral flow on any edge  $\{r, s\} \in E(G)$  is  $\lfloor \frac{1}{2} \rfloor = 0$  and the load-vector remains unchanged. The claim follows.  $\square$

In the remainder of this section we present two lower bounds for the deviation between the randomized rounding diffusion algorithm and the idealized process. First, we prove a bound of  $\Omega(\log n)$  for the hypercube. Together with Theorem 4.2 this implies that on hypercubes the quasirandom approach is as good as the randomized rounding diffusion algorithm.

**THEOREM 6.2.** *There is an initial load vector of the  $d$ -dimensional hypercube with  $n = 2^d$  vertices such that the deviation of the randomized rounding diffusion algorithm and the idealized process is at least  $\log n/4$  with probability  $1 - n^{-\Omega(1)}$ .*

*Proof.* We define an initial load vector  $x^{(0)}$  as follows. For every vertex  $v = (v_1, v_2, \dots, v_d) \in \{0, 1\}^d$  with  $v_1 = 0$  we set  $x_v^{(0)} = \xi_v^{(0)} = 0$  and if  $v_1 = 1$  we set  $x_v^{(0)} = \xi_v^{(0)} = d$ . Hence, the idealized process will send a flow of  $d/(2d) = 1/2$  from

every vertex  $v = (1, v_2, v_3, \dots, v_d) \in \{0, 1\}^d$  to  $(0, v_2, v_3, \dots, v_d)$ . Hence for the idealized process,  $\xi_v^{(1)} = (1/2)d$ , that is, all vertices have a load of  $(1/2)d$  after one step and the load is perfectly balanced.

Let us now consider the discrete process. Let  $V_0$  be the set of vertices whose bitstring begins with 0. Consider any node  $v \in V_0$ . Note that all neighbors of  $v$  have a load of 1 and the integral flow from any of those neighbors equals 1 with probability  $1/2$ , independently. Hence the load of  $v$  in the next step is just a binomial random variable and using the fact that  $\binom{r}{s} \geq (r/s)^s$  we obtain

$$\Pr \left[ x_v^{(1)} = \frac{3}{4}d \right] \geq \Pr \left[ x_v^{(1)} \geq \frac{3}{4}d \right] \geq \binom{d}{(3/4)d} 2^{-d} \geq \left( \frac{4}{3} \right)^{(3/4)d} 2^{-d} \geq n^{-1+C},$$

for some constant  $C > 0$  since  $d = \log_2 n$ . As the maximum degree of the graph is  $\log n$  and the size of  $V_0$  is  $n/2$ , it follows that there is a subset  $S \subseteq V_0$  of size  $\Omega(\frac{n}{\log^4 n})$  in the hypercube such that every pair in  $S$  has distance at least 4. By construction, the respective events  $x_v^{(1)} \geq (3/4)d$  are independent for all vertices  $v \in S$ . Hence

$$\Pr \left[ \exists v \in S: x_v^{(1)} \geq \frac{3}{4}d \right] \geq 1 - (1 - n^{-1+C})^{\Omega(\frac{n}{\log^4 n})} \geq 1 - n^{-C'},$$

where  $0 < C' < C$  is another constant. This means that with probability at least  $1 - n^{-C'}$  the load at vertex  $u$  at step 1 will be at least  $(3/4)d$  in the discrete process, but equals  $(1/2)d$  in the idealized process. This completes the proof.  $\square$

It remains to give a lower bound for the deviation between the randomized rounding and the idealized process for torus graphs. The following theorem proves a polylogarithmic lower bound for the randomized rounding algorithm which should be compared to the constant upper bound for the quasirandom approach of Theorem 5.4. Similar results can also be derived for non-uniform torus graphs.

**THEOREM 6.3.** *There is an initial load vector of the  $d$ -dimensional uniform torus graph with  $n$  vertices such that the deviation between the randomized rounding diffusion algorithm and the idealized process is  $\Omega(\text{polylog}(n))$  with probability  $1 - o(1)$ .*

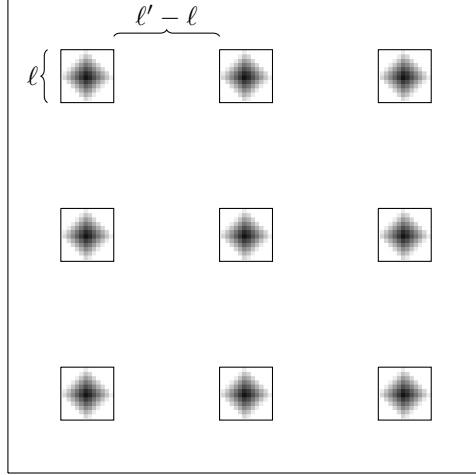
*Proof.* Let  $n$  be a sufficiently large integer and  $\mathbb{T}$  be a  $d$ -dimensional torus graph with  $n$  vertices and side-length  $\sqrt[d]{n} \in \mathbb{N}$ . Let  $B_k(u) := \{v \in V: \|v - u\|_\infty \leq k\}$  and  $\partial B_k(u) := \{v \in V: \|v - u\|_\infty = k\}$ . For every vertex  $v \in V(\mathbb{T})$ , we define  $|B_{\ell/2}(v)| = \ell^d = (\log n)^{1/4}$  with  $\ell := (\log n)^{1/(4d)}$ , where we assume w.l.o.g. that  $\ell$  is an odd integer. For  $\ell' := (\log n)^{2/(3d)}$ , define a set  $S \subseteq V$  by

$$S := \{(x_1 \ell', x_2 \ell', \dots, x_d \ell') \mid 1 \leq x_1, x_2, \dots, x_d < \sqrt[d]{n}/\ell' - 1\},$$

that is, every pair of distinct vertices in  $S$  has a coordinate-wise distance which is a multiple of  $\ell'$ . Note that  $|S| = \Omega(n/\ell'^d)$ . Define the initial load vector as  $x_i^{(0)} = \xi_i^{(0)} := 2d \cdot \max\{0, \ell/2 - \text{dist}(i, S)\}$ ,  $i \in V$ . Clearly, the initial discrepancy is  $K = 2d \cdot \ell/2$ .

The idea is now to decompose  $\mathbb{T}$  in smaller subgraphs centered around  $s \in S$ , since the upper bound on the convergence rate given by Theorem 3.1 has a strong dependence on the size of the graph. Then we relate the simultaneous convergence on each of the smaller graphs to the convergence on the original graph. An illustration of our decomposition of  $\mathbb{T}$  can be found in Figure 6.1.

Fix some  $s \in S$ . Then the subgraph induced by the vertices  $B_{\ell/2}(s)$  is a  $d$ -dimensional grid with exactly  $n' := (\log n)^{1/4}$  vertices. Let  $\mathbb{T}' = \mathbb{T}'(s)$  denote the



**Fig. 6.1:** Overview of the decomposition of  $\mathsf{T}$  into various  $\mathsf{T}'(s)$  for the two-dimensional case  $d = 2$ . The inner rectangles represent the various smaller grids  $\mathsf{T}'(s)$  with  $s \in S$ . The darkness indicates the amount of the initial load. Note that the initial load of vertices outside the  $\mathsf{T}'(s)$ 's is 0.

corresponding  $d$ -dimensional torus graph with the same vertices, but additional wrap-around edges between vertices of  $\partial B_{\ell/2}(s)$ . W.l.o.g. we assume that the side-length  $\sqrt[d]{n}$  of  $T$  is a multiple of the side-length  $\ell$  of  $\mathsf{T}'(s)$ . Let  $\mathbf{P}'$  be the diffusion matrix of  $\mathsf{T}'(s)$ .

Let us denote by  $\xi'^{(0)}$  ( $x'^{(0)}$ ) the projection of the load vector  $\xi^{(0)}$  ( $x^{(0)}$ ) from  $\mathsf{T}$  onto  $\mathsf{T}'(s)$ . By Corollary 3.2, the idealized process reduces the discrepancy on  $\mathsf{T}'(s)$  from  $K = (\log n)^{1/(4d)}/2$  to 1 within  $t_0 := \mathcal{O}((n')^{2/d} \log(Kn')) = \mathcal{O}(\log \log(n) (\log n)^{1/(2d)})$  time steps. We now want to argue that this also happens on the original graph  $\mathsf{T}$  with  $n$  vertices. Note that the convergence of the idealized process on  $\mathsf{T}'(s)$  implies

$$\|\xi'^{(t_0)} - \bar{\xi}'\|_\infty = \|\mathbf{P}'^{t_0} \xi'^{(0)} - \bar{\xi}'\|_\infty \leq 1. \quad (6.1)$$

Furthermore, note that the average load  $\bar{\xi}'$  in each  $\mathsf{T}'(s)$  satisfies

$$\bar{\xi}' \leq 2d \cdot \ell/4.$$

Our next observation is that for any two vertices  $u, v \in \mathsf{T}'(s)$ ,

$$\mathbf{P}_{u,v}^{t_0} \leq \mathbf{P}_{u,v}^{t_0} \quad (6.2)$$

as a random walk on  $\mathsf{T}'(s)$  can be expressed as a projection of a random walk on  $\mathsf{T}$  (by assigning each vertex in  $\mathsf{T}'(s)$  to a set of vertices in  $\mathsf{T}$ ). With the observations

- for  $v \in \mathsf{T}'(s)$ :  $\xi_v^{(0)} = \xi'_v{}^{(0)}$ ,
- for  $v \in \mathsf{T}$  and  $\ell/2 \leq \text{dist}(v, s) \leq t_0$ :  $\xi_v^{(0)} = 0$  (as  $t_0 = o(\ell' - \ell/2)$ ),
- for  $v \in \mathsf{T}$  and  $\text{dist}(v, s) > t_0$ :  $\mathbf{P}_{v,s}^{t_0} = 0$ ,

we obtain for any vertex  $v \in B_{\ell/2}(s)$ ,

$$\xi_s^{(t_0)} = (\mathbf{P}^{(t_0)} \cdot \xi^{(0)})_s = \sum_{v \in \mathsf{T}} \xi_v^{(0)} \mathbf{P}_{v,s}^{(t_0)} = \sum_{v \in \mathsf{T}'(s)} \xi'_v{}^{(0)} \mathbf{P}_{v,s}^{(t_0)}.$$

By first applying equation (6.2) and then equation (6.1), we get

$$\xi_s^{(t_0)} \leq \sum_{v \in \mathsf{T}'(s)} \xi'_v{}^{(0)} \mathbf{P}_{v,s}^{(t_0)} = \xi_s'^{(t_0)} \leq \bar{\xi}' + 1.$$

This means that the idealized process achieves after  $t_0$  time steps a good balancing at  $s$ . On the other hand, the discrete process may fail within  $t_0$  time steps if there is an  $s$  such that all edges in  $\mathsf{T}'(s)$  round towards  $s$  at all time steps  $t \leq t_0$ . (Note that by construction, no load from another  $\mathsf{T}'(s')$ ,  $s' \in S \setminus \{s\}$ , can reach  $\mathsf{T}'(s)$  within  $t_0$  steps, since the distance between any vertex in  $\mathsf{T}'(s)$  and  $\mathsf{T}'(s')$  is  $\ell' - 2\ell \geq t_0$ .) Moreover, by definition of  $x^{(0)}$ ,  $|x_u^{(0)} - x_v^{(0)}| \in \{0, 2d\}$  if  $\{u, v\} \in E(\mathsf{T})$ . Hence the fractional flow in the first step is  $\in \{0, \frac{1}{2}\}$  and for fixed  $s$  the probability that  $x_u^{(0)} = x_u^{(1)}$  for all  $u \in \mathsf{T}'(s)$  is at least  $2^{-|B_{\ell/2}(s)|}$ . By induction, for fixed  $s$  the probability that  $x_u^{(0)} = x_u^{(t_0)}$  holds for all  $u \in \mathsf{T}'(s)$  is at least

$$2^{-|B_{\ell/2}(s)| t_0} = 2^{-(\log n)^{1/4} \cdot \mathcal{O}(\log \log n) (\log n)^{1/(2d)}} \geq 2^{-(\log n)^{4/5}}.$$

As we have  $|S| = \Omega(n/\ell'^d) = \Omega(\text{poly}(n))$  independent events, it follows that there is at least one  $s \in S$  with  $x_s^{(t_0)} = x_s^{(0)} = \ell/2 \cdot 2d$  with probability

$$1 - \left(1 - 2^{-(\log n)^{4/5}}\right)^{\Omega(\text{poly}(n))} \geq 1 - n^{-C},$$

where  $C > 0$  is some constant. If this happens, then the deviation between the discrete and idealized process at vertex  $s \in S$  at step  $t_0$  is at least

$$|x_s^{(t_0)} - \xi_s^{(t_0)}| \geq |2d \cdot \ell/2 - (2d \cdot \ell/4 + 1)| = \Omega((\log n)^{1/(4d)}),$$

and the claim follows.  $\square$

**7. Conclusions.** We propose and analyze a new deterministic algorithm for balancing indivisible tokens. By achieving a constant discrepancy in optimal time on all torus graphs, our algorithm improves upon all previous deterministic and random approaches with respect to both running time and discrepancy. For hypercubes we prove a discrepancy of  $\Theta(\log n)$  which is also significantly better than the (deterministic) RSW-algorithm which achieves a discrepancy of  $\Omega(\log^2 n)$ .

On a concrete level, it would be interesting to extend these results to other network topologies. From a higher perspective, our new algorithm provides a striking example of quasirandomness in algorithmics. Devising and analyzing similar algorithms for other tasks such as routing, scheduling or synchronization remains an interesting open problem.

**Acknowledgments.** This work was done while all three authors were postdoctoral fellows at the International Computer Science Institute (ICSI) in Berkeley, California supported by the German Academic Exchange Service (DAAD). We also want to thank Benjamin Doerr for suggesting a simplified proof of Lemma 3.6.

## References.

- [1] W. Aiello, B. Awerbuch, B. M. Maggs, and S. Rao. Approximate load balancing on dynamic and asynchronous networks. In *25th Annual ACM Symposium on Theory of Computing (STOC'93)*, pp. 632–641, 1993.
- [2] R. D. Barve, E. F. Grove, and J. S. Vitter. Simple randomized mergesort on parallel disks. *Parallel Computing*, 23:601–631, 1997.
- [3] F. Chung. *Spectral Graph Theory*. AMS, revised edition, 2006. URL <http://www.math.ucsd.edu/~fan/research/revised.html>.
- [4] J. Cooper and J. Spencer. Simulating a random walk with constant error. *Combinatorics, Probability & Computing*, 15:815–822, 2006.
- [5] G. Cybenko. Load balancing for distributed memory multiprocessors. *Journal of Parallel and Distributed Computing*, 7:279–301, 1989.

- [6] P. Diaconis, R. Graham, and J. Morrison. Asymptotic analysis of a random walk on a hypercube with many dimensions. *Random Structures and Algorithms*, 1:51–72, 1990.
- [7] B. Doerr, T. Friedrich, and T. Sauerwald. Quasirandom rumor spreading. In *19th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'08)*, pp. 773–781, 2008.
- [8] R. Elsässer and T. Sauerwald. Discrete Load Balancing is (Almost) as Easy as Continuous Load Balancing. In *29th Annual ACM Principles of Distributed Computing (PODC'10)*, 2010. to appear.
- [9] R. Elsässer, B. Monien, and S. Schamberger. Distributing unit size workload packages in heterogenous networks. *Journal of Graph Algorithms & Applications*, 10: 51–68, 2006.
- [10] J. A. Fill. The passage time distribution for a birth-and-death chain: Strong stationary duality gives a first stochastic proof. *Journal of Theoretical Probability*, 22: 543–557, 2009.
- [11] T. Friedrich and T. Sauerwald. Near-perfect load balancing by randomized rounding. In *41st Annual ACM Symposium on Theory of Computing (STOC'09)*, pp. 121–130, 2009.
- [12] T. Friedrich, M. Gairing, and T. Sauerwald. Quasirandom load balancing. In *21st Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'10)*, pp. 1620–1629. SIAM, 2010.
- [13] J. Gehrke, C. Plaxton, and R. Rajaraman. Rapid convergence of a local load balancing algorithm for asynchronous rings. *Theoretical Computer Science*, 220:247–265, 1999.
- [14] B. Ghosh and S. Muthukrishnan. Dynamic load balancing by random matchings. *Journal of Computer and System Sciences*, 53:357–370, 1996.
- [15] B. Ghosh, F. T. Leighton, B. M. Maggs, S. Muthukrishnan, C. G. Plaxton, R. Rajaraman, A. W. Richa, R. E. Tarjan, and D. Zuckerman. Tight analyses of two local load balancing algorithms. *SIAM Journal on Computing*, 29:29–64, 1999.
- [16] G. Grimmet and D. Stirzaker. *Probability and Random Processes*. Oxford University Press, 3rd edition, 2001.
- [17] V. Guruswami. Rapidly Mixing Markov Chains, 2000. unpublished, available at <http://www.cs.washington.edu/homes/venkat/pubs/pubs.html>.
- [18] S. G. Hoggar. Chromatic polynomials and logarithmic concavity. *Journal of Combinatorial Theory, Series B*, 16:248 – 254, 1974.
- [19] G. Jan and Y. Hwang. An efficient algorithm for perfect load balancing on hypercube multiprocessors. *The Journal of Supercomputing*, 25:5–15, 2003.
- [20] S. Karlin and J. McGregor. Coincidence properties of birth and death processes. *Pacific J. Math.*, 9:1109–1140, 1959.
- [21] S. Karlin, B. Lindqvist, and Y.-C. Yao. Markov chains on hypercubes: Spectral representations and several majorization relations. *Random Structures & Algorithms*, 4:1–36, 1993.
- [22] J. Keilson. *Markov Chain Models – Rarity and Exponentiality*. Springer Verlag, 1979.
- [23] J. Keilson and H. Gerber. Some results for discrete unimodality. *Journal of the American Statistical Association*, 66:386–389, 1971.
- [24] G. F. Lawler. *Intersections of random walks*. Probability and its Applications. Birkhäuser, 1991.
- [25] G. F. Lawler and V. Limic. *Random Walk: A Modern Introduction*. Cambridge Studies in Advanced Mathematics. Cambridge University Press, 2010.
- [26] L. Lovász. Random walks on graphs: A survey. *Combinatorics, Paul Erdős is*

- Eighty*, 2:1–46, 1993.
- [27] L. Lovász and P. Winkler. Mixing of random walks and other diffusions on a graph. *Surveys in combinatorics*, pp. 119–154, 1995.
  - [28] B. Morris and A. Sinclair. Random walks on truncated cubes and sampling 0-1 knapsack solutions. *SIAM Journal on Computing*, 34:195–226, 2004.
  - [29] R. Motwani and P. Raghavan. *Randomized Algorithms*. Cambridge University Press, 7th edition, 1995.
  - [30] S. Muthukrishnan, B. Ghosh, and M. H. Schultz. First- and second-order diffusive methods for rapid, coarse, distributed load balancing. *Theory of Computing Systems*, 31:331–354, 1998.
  - [31] C. Plaxton. Load balancing, selection sorting on the hypercube. In *1st ACM Symposium on Parallel Algorithms and Architectures (SPAA '89)*, pp. 64–73, 1989.
  - [32] Y. Rabani, A. Sinclair, and R. Wanka. Local divergence of Markov chains and the analysis of iterative load balancing schemes. In *39th Annual IEEE Symposium on Foundations of Computer Science (FOCS'98)*, pp. 694–705, 1998.
  - [33] S. Ross. *Introduction to Probability Models*. Academic Press, 2007.
  - [34] R. P. Stanley. Log-concave and unimodal sequences in algebra, combinatorics, and geometry. *New York Academy Sciences Annals*, 576:500–535, 1989.
  - [35] R. Subramanian and I. D. Scherson. An analysis of diffusive load-balancing. In *6th ACM Symposium on Parallel Algorithms and Architectures (SPAA '94)*, pp. 220–225, New York, NY, USA, 1994. ACM.
  - [36] S. Surana, B. Godfrey, K. Lakshminarayanan, R. Karp, and I. Stoica. Load balancing in dynamic structured peer-to-peer systems. *Performance Evaluation*, 63: 217–240, 2006.
  - [37] R. D. Williams. Performance of dynamic load balancing algorithms for unstructured mesh calculations. *Concurrency: Practice and Experience*, 3:457–481, 1991.
  - [38] D. Zhanga, C. Jianga, and S. Li. A fast adaptive load balancing method for parallel particle-based simulations. *Simulation Modelling Practice and Theory*, 17:1032–1042, 2009.

#### Appendix A. Proof of a supplementary result.

In order to prove Proposition 4.10, we first note the following elementary lemma.

LEMMA A.1. *Let  $(a_k)_{k=1}^d, (b_k)_{k=1}^d, (c_k)_{k=1}^d$  be three positive sequences such that*

- (i) for all  $j \in [1, d]$ ,  $\sum_{k=j}^d a_k \leq \sum_{k=j}^d b_k$ ,*
- (ii)  $c_k$  is monotone increasing in  $k$ .*

*Then for all  $j \in [1, d]$ ,  $\sum_{k=j}^d a_k \cdot c_k \leq \sum_{k=j}^d b_k \cdot c_k$ .*

*Proof.* Define  $i := d - j + 1$ . We will show that

$$\sum_{k=d-i+1}^d a_k \cdot c_k \leq \sum_{k=d-i+1}^d b_k \cdot c_k \quad (\text{A.1})$$

for all  $i \in [1, d]$ . Our proof is by induction on the number of summands  $i \in [1, d]$ . The claim is trivial for  $i = 1$  and  $i = 2$ . Assume inductively that (A.1) holds for all  $i \geq i'$  and for all sequences satisfying the conditions of the lemma. We will show the claim for  $i = i' + 1$ , i.e.,

$$\sum_{k=d-i'}^d a_k \cdot c_k \leq \sum_{k=d-i'}^d b_k \cdot c_k.$$

Define two shorter sequences  $(a'_k)_{k=1}^{d-1}$  and  $(b'_k)_{k=1}^{d-1}$  as follows:

- $a'_k = a_k$  for  $k < d-1$  and  $a'_{d-1} := a_{d-1} + \frac{c_d}{c_{d-1}}a_d$
- $b'_k = b_k$  for  $k < d-1$  and  $b'_{d-1} := b_{d-1} + \frac{c_d}{c_{d-1}}b_d$

We will show that the sequences  $(a'_k)_{k=1}^{d-1}, (b'_k)_{k=1}^{d-1}, (c_k)_{k=1}^{d-1}$  satisfy the conditions of the lemma. Since  $(c_k)_k$  remained unchanged it suffices to show that for all  $j' \in [1, d-1]$ ,

$$\sum_{k=j'}^{d-1} a'_k \leq \sum_{k=j'}^{d-1} b'_k,$$

or equivalently (using the definition of  $a'_k$  and  $b'_k$ )

$$\sum_{k=j'}^d a_k + \left( \frac{c_d}{c_{d-1}} - 1 \right) a_d \leq \sum_{k=j'}^d b_k + \left( \frac{c_d}{c_{d-1}} - 1 \right) b_d.$$

By the first assumption of the lemma we have

$$\sum_{k=j'}^d a_k \leq \sum_{k=j'}^d b_k.$$

Moreover, since  $a_d \leq b_d$  and  $\frac{c_d}{c_{d-1}} \geq 1$ , we have

$$\left( \frac{c_d}{c_{d-1}} - 1 \right) a_d \leq \left( \frac{c_d}{c_{d-1}} - 1 \right) b_d.$$

Thus,  $(a'_k)_{k=1}^{d-1}, (b'_k)_{k=1}^{d-1}, (c_k)_{k=1}^{d-1}$  satisfy the conditions of the lemma. By induction hypothesis on those sequences and for  $i'$  summands, we have

$$\sum_{k=d-i'}^{d-1} a'_k \cdot c_k \leq \sum_{k=d-i'}^{d-1} b'_k \cdot c_k.$$

Plugging in the definition of  $a'_k$  and  $b'_k$  we finally obtain

$$\sum_{k=d-i'}^{d-2} a_k \cdot c_k + c_{d-1} \left( a_{d-1} + \frac{c_d}{c_{d-1}} a_d \right) \leq \sum_{k=d-i'}^{d-2} b_k \cdot c_k + c_{d-1} \left( b_{d-1} + \frac{c_d}{c_{d-1}} b_d \right),$$

which is precisely the induction claim for  $i' + 1$ . The lemma follows.  $\square$

**Proposition 4.10.** *Let  $i, j \in V$  be two vertices of the  $d$ -dimensional hypercube with  $\text{dist}(i, j) \geq d/2$ . Then,  $\mathbf{P}_{i,j}(t)$  is monotone increasing.*

*Proof.* Fix an arbitrary step  $t \in \mathbb{N}$ . By symmetry, it suffices to prove that  $\mathbf{P}_{0,x}(t) \leq \mathbf{P}_{0,x}(t+1)$  where  $x \in \{0,1\}^d$  with  $|x| \geq d/2$ . First note that for all  $j \in [|x|, d]$ ,  $\sum_{k=j}^d \mathbf{Pr}[\text{exactly } k \text{ coordinates chosen in } t \text{ steps}] \leq \sum_{k=j}^d \mathbf{Pr}[\text{exactly } k \text{ coordinates chosen in } t+1 \text{ steps}]$  since the distribution of chosen coordinates after  $t+1$  steps clearly dominates the distribution of chosen coordinates after  $t$  steps. Observe that for any  $|x| \geq d/2$  the function  $f(k) := 2^{-k} \binom{d-|x|}{k-|x|} / \binom{d}{k}$  is monotone increasing in  $|x| \leq k \leq d$ . This can be verified by showing that  $f(k)/f(k-1) \geq 1$  for any  $k$  with  $|x| < k \leq d$ . This allows us to apply Lemma A.1 to equation (4.6), giving that  $\mathbf{P}_{0,x}(t) \leq \mathbf{P}_{0,x}(t+1)$ . Hence the proposition follows.  $\square$